ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ
ΒΙΟΙΑΤΡΙΚΗ»

Μέθοδοι Βαθμονόμησης Κάμερας για εφαρμογές Υπολογιστικής Όρασης

Κότταρη Κωνσταντίνα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιστημονικός Υπεύθυνος

Δελήμπασης Κωνσταντίνος

Επίκουρος Καθηγητής

Λαμία, 2016

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

Interdepartmental Postgraduate Master's Program in
«INFORMATICS AND COMPUTATIONAL BIOMEDICINE»

Camera Calibration Methods for Computer Vision Applications

Kottari Konstantina

MSc Thesis

Scientific Supervisor:

Delibasis Konstantinos

Assistant Professor

Lamia, 2016

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Λαμία, 2016

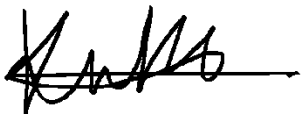
«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «*Μέθοδοι Βαθμονόμησης Κάμερας για εφαρμογές Υπολογιστικής Όρασης*» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Η ΔΗΛΟΥΣΑ

ΚΟΤΤΑΡΗ ΚΩΝΣΤΑΝΤΙΝΑ

Ημερομηνία: *29/10/2016*



Υπογραφή

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Camera Calibration Methods for Computer Vision Applications

Kottari Konstantina

Examination committee:

Delibasis Konstantinos – Assistant Professor

Plagianakos Vassilis – Associate Professor

Iakovidis Dimitrios – Associate Professor

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Abstract

Video processing and analysis applications are part of Artificial Intelligence. Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. It has been studied extensively in computer vision and photogrammetry, and even recently new techniques have been proposed. In this work, we first review thoroughly two techniques of perspective calibration [1], [2] within a consistent mathematical framework, using both linear and non-linear estimation of the cameras' projection matrices. Further, we propose some mathematical variations on these approaches. The calibration of perspective camera has been implemented using Matlab. Then, two computer vision applications were selected to be implemented and tested, by utilizing the calibration results of a number of available cameras: real-time position estimation, as well as volumetric reconstruction from silhouettes.

Results are obtained for an experimental setup of three very low cost projective IP cameras. Camera calibration results are presented for both linear and non-linear methods, measuring the accuracy of re-projection of salient points with known positions. Applications' results are presented in reconstructing or estimating the position of synthetic 3D human models, as well as real human subjects. The accuracy of position estimation was measured for points on the floor, as well as small objects and humans. Finally the accuracy of volumetric reconstruction was measured using a simple object with known geometry. The effect of calibration error on the accuracy of volumetric reconstruction is also quantified.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Keywords

Artificial Intelligence, Video Processing, Projective Camera, Camera Calibration, Silhouette Detection, Silhouette Segmentation, Position Estimation, Silhouette to Shape/Shape from Silhouettes, Three-dimensional Human Modelling, Volumetric Model, Calibration Effect on Reconstruction.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Acknowledgement

I would like to express my gratitude to my supervisor Kostas Delibasis for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore, I would like to thank my loved ones, who have supported me throughout the entire process, both by keeping me harmonious and helping me putting pieces together. Finally, I would like to dedicate this work to my beloved grandmother, who wanted me to become more and more active and successful in science and research and gives me power even without being here. I will be grateful forever for your knowledge and love.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Contents

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

1.Introduction

1.1. Calibration and Projective Geometry

The formal definition of calibration by the International Bureau of Weights and Measures is the following: "Operation that, under specified conditions, in a first step, establishes a relation between the quantity values with measurement uncertainties provided by measurement standards and corresponding indications with associated measurement uncertainties (of the calibrated instrument or secondary standard) and, in a second step, uses this information to establish a relation for obtaining a measurement result from an indication."

Calibration, in general, is the process of finding a relationship between two quantities that are unknown (when the measurable quantities are not given a particular value for the amount considered or found a standard for the quantity). When one quantity is known, which is made or set with one device, another measurement is made as similar way as possible with the first device using a second device. The measurable quantities may differ in two devices which are equivalent. The device with the known or assigned correctness is called the standard. The second device is the unit under test, test instrument, or any of several other names for the device being calibrated.

Calibration of an instrument is the process of finding a relationship between the measurements of this instrument and the measurements of a standard instrument of the same type, whose measurements are supposed to be correct. For instance, thermometer may be calibrated so that its reading corresponds to the correct temperature (eg. in Celsius). A fundamental way of performing that could be by marking the reading when in melting ice and in boiling water (by definition at 0 and 100 Celcius respectively). The range in-between these two readings is divided into 100 steps, each corresponding to 1 degree Celsius. Alternatively, the thermometer can be calibrated against another, already calibrated thermometer.

In computer vision, camera calibration is defined as the recovery of the transformation between points in real world and image pixels. The transformations needed are derived from projective geometry, using known positions of convenient points or geometric patterns. These transformations include geometric transformations between the real world and the camera coordinate system, as well as between the camera sensor and the pixel coordinate system.

The study of projective geometry was initiated by the painters of the Italian Renaissance, who wanted to produce a convincing illusion of 3D depth in their

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

architectural paintings [3]. They made considerable use of vanishing points and derived several practically useful geometric constructions, for example to split a projected square into four equal sub-squares, or to find the projection of a parallelogram when the projections of the two of its sides are known.

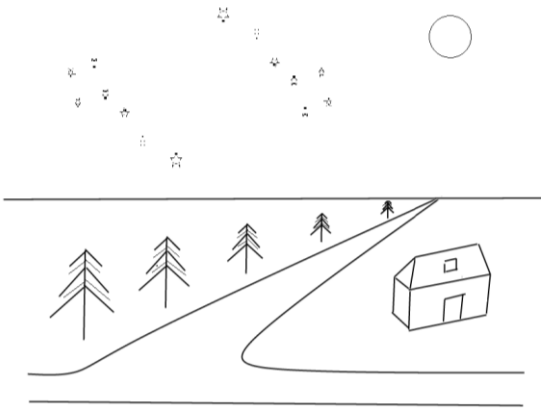


Figure 1. Simple painting using projective geometry.

The edges of the road of Fig.1 are parallel lines in 3D space, but in the image they appear to converge as they recede towards the horizon. The line of the horizon is formed by the "infinitely distant points" or vanishing directions of the ground plane. Any pair of parallel, horizontal lines appears to meet at the point of the horizon corresponding to their common direction. This is true even if they lie at different heights above the ground plane. Moreover, any two horizontal planes appear to come together in the distance, and intersect in the horizon line or "line at infinity". All of these "intersections at infinity" stay constant as the observer moves. The road always seems to disappear at the same point (direction) on the horizon, and the stars stay fixed as you walk along: lines of sight to infinitely distant points are always parallel, because they "(only) meet at infinity". These simple examples show the effect of central projection image formation model and can be used in camera calibration.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

1.2. The Perspective Projection Camera Model

The perspective camera model consists of the charged-coupled device (CCD) image sensor, which is perpendicular to the optical axis of the camera, which by convention is assumed to be the *Z* axis. The center of the sensor is called the principle point. The distance between the sensor and the center of projection is called principle distance or focal length of the camera. The details of the camera model are shown in Fig.2.

The perspective projection camera model is based on central projection and the center of projection is the origin of the camera coordinate system. Thus, a point **P** of the real world (scene) will be projected at the position where the line segment between the point **P** and the center of projection intersects the sensor ($\mathbf{P_c}$) – without taking into account image distortion.

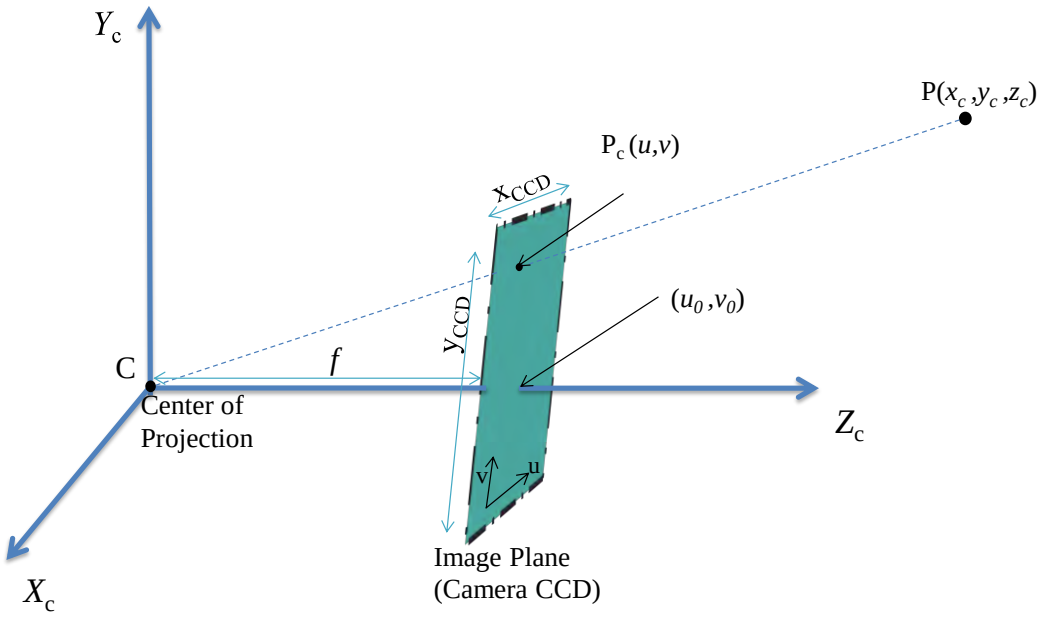


Figure 2. The perspective projection camera model. The principle distance *f*, the center of projection and a point P of the scene are plotted in the coordinate system of the camera. The coordinates of the principal point (u_0, v_0) and the projection P_c of point P are plotted on the image plane of the camera sensor.

Basic camera calibration is the recovery of the principle distance *f* (the focal length of the camera) and the principle point (u_0,v_0) (the center of the camera sensor) in the image plane – image coordinate system. This is referred to as *interior* orientation in photogrammetry and the focal length and the center of the camera sensor as intrinsic parameters.

A calibration target can be imaged to provide correspondences between points in the image and points in space. The relationship between the target coordinate system and the camera coordinate system typically needs to be recovered from the correspondences. This is referred to as *exterior* orientation in photogrammetry.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Exterior Orientation (external camera parameters) is the relationship between a scene-centered coordinate system and a camera-centered coordinate system. External camera parameters define the transformation (*M*) from scene to camera consists of a rotation (*R*) and a translation (*T*), where:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

The inverse transformation $M^{-1} = T^{-1} \cdot R^{-1}$ is shown in Fig.3. The rotation matrix *R* can be written using trigonometric functions with only three degrees of freedom (dof). However, when using Minimum Linear Least Squares optimization to recover the calibration parameters, the formalism of Eq.(1) is used, which leads to 9 dof (not independent though). Translation matrix T has 3 degrees of freedom, irrespectively of the optimization method used. Therefore, transformation M may have 12 dof, nine for rotation and three for translation.

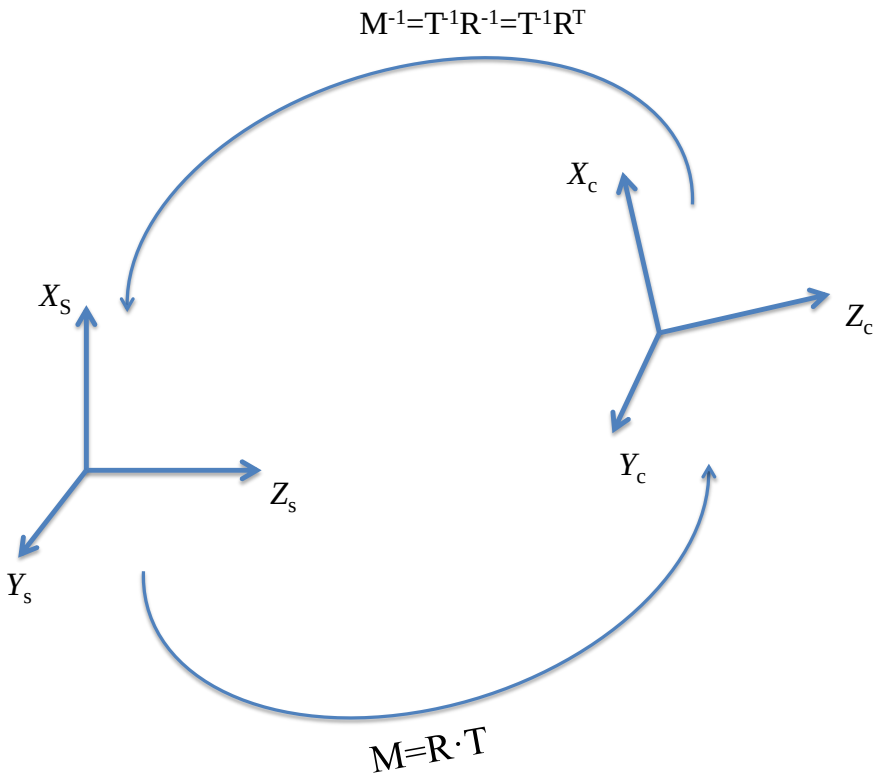


Figure 3. The transformation from a scene coordinate system to a camera-centered coordinate system and vice versa. The transformation matrix *M* from scene to camera

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

consists of rotation (*R*) translation (*T*). The opposite transformation is the inverse matrix M^{-1}.

The scene coordinate system can be any system convenient for the particular experimental setup. In the case of a planar target, the Z axis is usually chosen perpendicular to the plane, with $Z = 0$ in the target plane.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

1.3. Computer Vision Applications

The field of automated human activity recognition utilizing fixed cameras of indoor environments has gained significant interest during the last years. It finds a variety of applications in diverse areas, such as assistive environments, smart homes, support for the elderly or the chronic ill, surveillance and security, traffic control, industrial processes, etc. Automatic estimation of the position of an object in video sequences, or recovery of the volumetric representation of a real object by utilizing silhouettes from multiple cameras (silhouette to shape algorithm) are essential application steps for the aforementioned goals.

1.3.1. *Position Estimation*

The proposed algorithm is able to estimate the position of the human silhouette, by the knowledge of both intrinsic and extrinsic parameters of the cameras, since they are previously calibrated. The position of the object can be estimated every time a single camera captures it. Moreover, the concurrent use of multiple cameras improves the accuracy of the estimation.

1.3.2. *Silhouette to Shape*

Using silhouettes to get shapes is an active topic in computer vision. Recovering 3D structure (shape) from multiple binary (2D) silhouettes (silhouette to shape – S2S, commonly referred as shape from silhouette -SfS) by projective cameras is a well-known technique [4]. In [5] the concept of visual hull, as the maximal volume that would produce the available 2D silhouettes, is studied. Space carving is a technique of constructing the visual hall by volume elements of the real 3D space that are projected inside all available 2D silhouettes. An alternative technique for visual hull formation in the form of a triangulated surface is the "Marching intersections", is described in [6].

In the space carving algorithm, a single silhouette image of an object that can be acquired by a single camera frame would be combined with the camera parameters of the calibration and may, then, be used to back-project the silhouette area. It is known that the 3D object lies inside the volume generated by this back-projection. With multiple views of the same object, we can intersect the *generalized cones* generated by each image, to build a volume which is guaranteed to contain the object. The limiting smallest volume obtainable in this way is known as the *visual* or *optic hull* of the object.

If perfect calibration and segmentation is available, infinite views of an object will result to shape reconstruction identical with the real object, as long as it is *convex*.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2. Methodology

In this work, the algorithm for calibration of a perspective camera is analyzed. A linear, as well as a non-linear optimization is applied for the estimation of the cameras' projection matrices. Both algorithms are applied on three identical projective cameras, installed on the walls of our laboratory. Moreover, applications of estimation of subject's real world position and volumetric shape reconstruction are implemented, using images acquired by the three calibrated cameras.

More specifically, the steps of the work are the following (Fig.4). Firstly, the three identical pinhole (projective) cameras are calibrated. The intrinsic parameters are obtained from the manufacturer. Since the cameras are fixed on the walls of the laboratory, their position has been measured manually and assumed known for the calibration process. Then, frames containing an object/person are acquired by the three projective cameras. The frames undertake segmentation (by background subtraction). Using the segmented frames that contain the binary silhouette and the calibration of each camera, the real world position of the object is estimated and an implementation of the space carving algorithm (S2S algorithm), is applied to generate a volumetric model of the object.

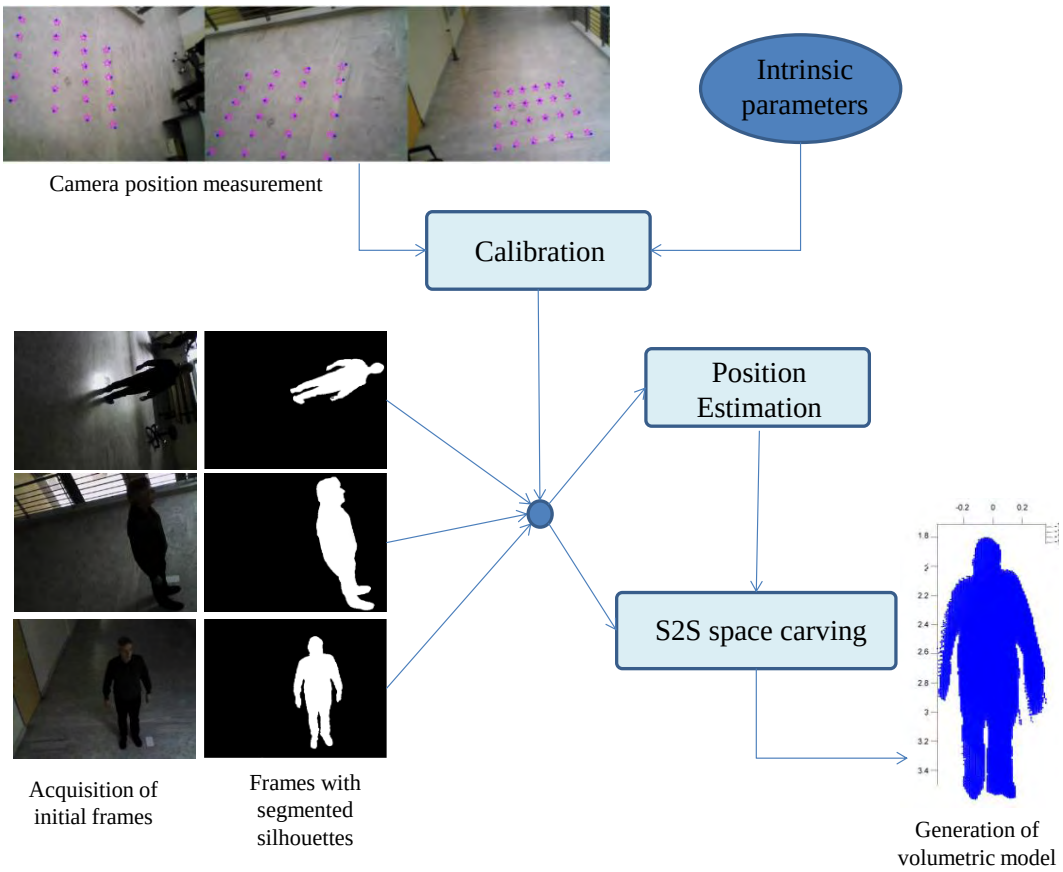


Figure 4. The steps of this work.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2.1. Homogeneous Coordinates

Every point in an image represents a possible line of sight of an incoming light ray: any 3D point along the ray is projected on the same image point, so only the direction of the ray is of interest, whereas the distance of the point along it cannot be recovered from mono-ocular view. In computer vision we need to represent this "virtual sphere" of incoming ray directions. This is commonly done by using "homogeneous" coordinates, defined as following: if the plane of projection lies at z=1, then any point (x_1,y_1) on this plane is the projection of $(x_1,y_1,1)$. The third homogeneous coordinate is chosen to be 1 for simplicity. In reality all points in 3D with (ax_1,ay_1,a) are projected on the same point. Thus, points in 2D plane are represented by their homogeneous triplet coordinates, whereas points (x,y,z) in 3D space are represented as $(x,y,z,1)$. This seems inefficient, but it has the significant advantage of making the image projection, as well as affine transformation process much easier to deal with. More specifically, it enables us to use matrix multiplication to express geometric transformation, even when they include translations. Therefore, all the matrices used in this work, unless otherwise stated, correspond to homogeneous coordinates.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2.2. Calibration Model

As it can be derived from the geometry of Fig.5, the camera calibration matrix that maps 3D point $\mathbf{p}_s$ to 2D point (u,v) can be found by using similar triangles as:

$$\frac{f}{z_c} = \frac{u}{x_c} = \frac{v}{y_c}$$

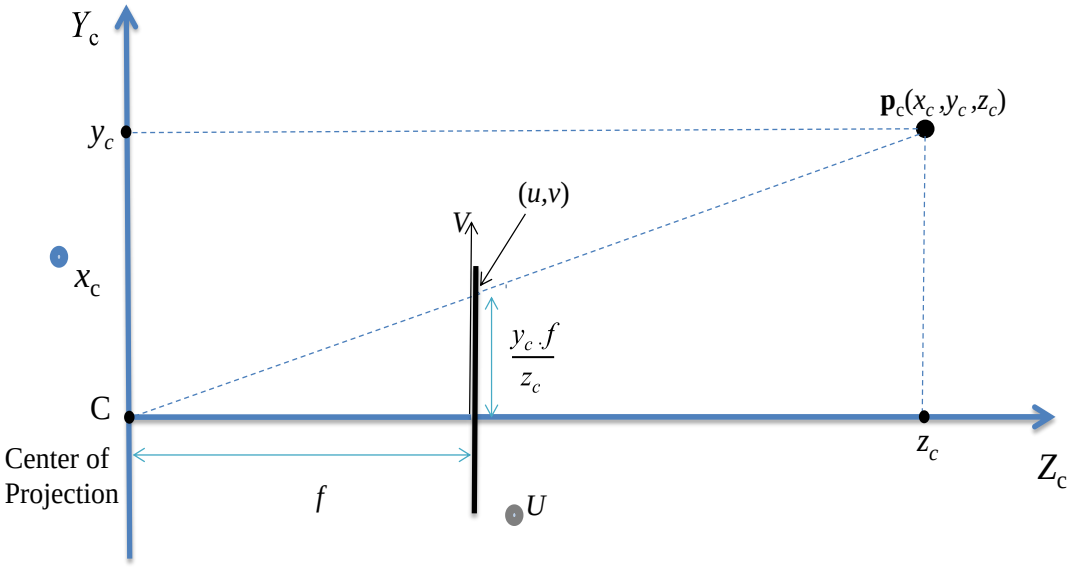


Figure 5. The geometry of the perspective projection camera model used for calibration. The principle distance *f*, the center of projection and a point P of the scene are plotted in the coordinate system of the camera. The coordinates of the projection P_c of point P are plotted on the image plane of the camera sensor. The similar triangles are the (small) triangle formed by points *C*,(0,0,*f*), (*u*,*v*) and the triangle formed by points *C*, (0,0,z_c), p_c.

In [1], the relationship between the coordinates of a real world (scene) point ($\mathbf{p}_s$) in the camera frame of reference $\mathbf{p}_c = (x_c, y_c, z_c)$ and the sensor coordinates are calculated using the perspective projection equations:

$$\frac{u}{f} = \frac{x_c}{z_c}, \quad \frac{v}{f} = \frac{y_c}{z_c} \qquad 3$$

where f is the focal length of the camera (Fig.5) and (u,v) are the coordinates of the projection of $\mathbf{p}_c$ on the image sensor measured in units of length. By solving Eq.(3) for (u,v) we obtain:

$$u = \frac{x_c \cdot f}{z_c}, \; v = \frac{y_c \cdot f}{z_c} \qquad 4$$

This is the equation of projection and can be expressed in matrix form by using homogeneous coordinates as:

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

$$\begin{pmatrix} u \\ v \\ o \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \quad (5)$$

from where the point $(u,v,o)=\left(\frac{x_c \cdot f}{z_c}, \frac{y_c \cdot f}{z_c}, 1\right)$ is generated.

A complicating factor in the calibration of many modern electronic cameras is that the discrete nature of image sampling is not preserved in the signal. This unknown horizontal scale factor (s_u), as mentioned in [1], can be recovered as part of the camera calibration process via modifying Eq.(4) as following:

$$u = s_u \frac{x_c \cdot f}{z_c},\ v = \frac{y_c \cdot f}{z_c} \quad (6)$$

In this work we assume that s_u is equal to 1.

2.2.1. *Intrinsic camera parameters*

Interior Orientation is the relationship between sensor coordinates and image coordinates. The camera coordinate system has its origin at the center of projection, its *Z* axis along the optical axis, and its *X* and *Y* axes parallel to the *X* and *Y* axes of the image, as shown in Fig.5.

The problem in determining intrinsic camera parameters is the recovery of the principle point (u_0,v_0) and the principle distance *f* (Intrinsic Parameters). This is the basic task of camera calibration. However, in practice we also need to recover the position and attitude of the calibration target in the camera coordinate system.

In this work, the known parameters are only the size of the sensor (x_{CCD} and y_{CCD}), as they are indicated in Fig.5, and the number of rows and columns of the generated image ($[1, N_{col}]$ at *X* axis and $[1, N_{line}]$ at the *Y* axis). The recovery of the focal length will be discussed in the coplanar Section for salient world points $\mathbf{p}_{s,k} = [x_k, y_k, z_k]^T, k = 1,2,\ldots,N$ that are on the same plane. However, the affine transformation can be acquired even with the field of view (FoV) of the cameras being an unknown parameter.

Affine transformation is the mapping between the image plane and the image frame. The affine transformation from real image coordinates (*u*, *v*) to frame buffer (pixel) image coordinates (*j*, *i*) is the following pair of equations:

$$\begin{aligned} j &= au + c_1 \\ i &= bv + c_2 \end{aligned} \quad (7)$$

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Factors a and c_1 can be found by the knowledge that the first and last column of the image frame coordinates correspond to the extremes of the X axis of the real image $\left[-\frac{x_{CCD}}{2}, \frac{x_{CCD}}{2}\right]$, since the size of the CCD and of the image are known parameters:

$$\begin{aligned} a &= \frac{N_{col}-1}{x_{CCD}} \\ c_1 &= \frac{N_{col}+1}{2} \end{aligned} \quad (8)$$

Whereas factors b and c_2 can be found by the knowledge that the first and last line of the frame image coordinates correspond to the extremes of the Y axis of the sensor $\left[-\frac{y_{CCD}}{2}, \frac{y_{CCD}}{2}\right]$, as following:

$$\begin{aligned} b &= \frac{N_{line}-1}{y_{CCD}} \\ c_1 &= \frac{N_{line}+1}{2} \end{aligned} \quad (9)$$

Finally, using Eq.(8) and Eq.(9), the affine transformation of Eq.(7) becomes:

$$\begin{aligned} j &= \frac{N_{col}-1}{x_{CCD}} \cdot u + \frac{N_{col}+1}{2} \\ i &= \frac{N_{line}-1}{y_{CCD}} \cdot v + \frac{N_{line}+1}{2} \end{aligned} \quad (10)$$

2.2.2. *Extrinsic camera parameters*

Extrinsic parameters are the rotation (R) and the translation (T) of the transformation (M) from scene to camera coordinate system. Figure 3 depicts the forward and inverse transformation between the two coordinate systems. Let $\mathbf{p_s} = [x, y, z]^T$ and $\mathbf{p_c} = [x_c, y_c, z_c]^T$ be the column vectors of a point in real world (scene) and camera coordinate system, respectively. $\mathbf{p_s}$ and $\mathbf{p_c}$ are related by an orthonormal rotation matrix R and a translation matrix $T = [t_x, t_y, t_z]^T$. The relationship between the camera and the scene coordinate systems is given as:

$$\mathbf{p_c} = R \cdot \mathbf{p_s} + T \quad (11)$$

The aforementioned equation rewritten in matrix form becomes:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (12)$$

The above equation can be written in homogeneous coordinates, using Eq.(1), as:

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \\ 1 \end{pmatrix} \quad (13)$$

Equivalently, using also Eq.(2), equation Eq.(13) becomes:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (14)$$

The orthonormal rotation matrix R might be decomposed into Euler angles [7] or into a matrix of rotation around the vector of the camera optical axis $\mathbf{v}$.

Euler Angles

The rotation around Euler angles are defined as matrices around *X*, *Y* and *Z* axis in a sequential manner:

$$R = R_z \cdot R_x \cdot R_y \quad (15)$$

The rotation matrices for each axis are (without using homogeneous coordinate notation):

$$R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}, R_y = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \quad (16)$$

where θ_z, θ_x, θ_y are the angles of rotation around the corresponding axis.

However, representing 3D rotations using Euler angles has disadvantages such as dependence on the sequence of application of the rotations [8] and gimbal lock [9].

A sequence of rotations around Euler angles is implemented as the non-commutative matrix multiplication [10]. Thus, change of sequence of the rotation matrices affects respectively the result of the final rotation in the 3D space. This dependence renders successive rotations inefficient for 3D transformations, as well as counter-intuitive.

Let *x* represent a pure rotation around *X* axis, *y* represent a pure rotation around *Y* axis and *z* represent a pure rotation around *Z* axis. The order of 3D rotations could be represented as: *xyz*, *yzx*, *zxy* or reversing the order *zyx*, *xzy*, *yxz* – which gives 6 permutations. However, permutations may be more than six, because the angles are not independent. For instance rotating 90° around *X* axis, followed by 90° around *Y* axis and -90° around *X* axis, concludes to the same position as a single rotation of 90° around *Z* axis. Thus, any 3D rotation can be formed by combining rotations in just 2 planes.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

More specifically, if a sequence of rotations is applied once in a specific order and secondly in a changed order, then the result is different. For example, if we apply:

1. Rotation 90° around *X* axis
2. Rotation 90° around *Y* axis
3. Rotation -90° around *X* axis

the result is a 90 degrees rotation around *Z* axis. Whereas, if we apply:

1. Rotation 90° around *X* axis
2. Rotation -90° around *X* axis
3. Rotation 90° around *Y* axis

the result is a 90 degrees rotation around *Y* axis, since the first two rotations are negated.

Furthermore, the second axis of which rotation takes place cannot reach 90 degrees because that will cause the two other axes to align and the rotation around them will have effect on the same plane. This is the effect of gimbal lock.

Rotation around the Vector of the Camera Optical Axis

The rotation around vector is commonly implemented using the Rodrigues Rotation Formula [11], with the following rotation matrix (indicated without homogenous coordinates for complexity reasons):

$$R = I_{3\times3} + \sin\theta \cdot W + (1-\cos\theta)\cdot W^2 \ , \tag{17}$$

where $I_{3\times3}$ is the 3×3 unit matrix and *W* is the following antisymmetric matrix:

$$W = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{18}$$

However, the Rodrigues Rotation Formula has the disadvantage of generating a unit matrix in the case of rotating round a given vector by an angle equal to zero, irrespectively of the given vector. In order to overcome the aforementioned disadvantage, in this work, the rotation around vector is implemented as a combination of a rotation matrix M that matches the unit vector $\mathbf{c} = (c_x, c_y, c_z)$ of the camera optical axis with the *Z* axis and a rotation matrix R_z, around the *Z* axis:

$$R = R_z \cdot A \tag{19}$$

The R_z matrix is identical to the *Z* axis rotation matrix of Euler angles Eq.(16) and it can be proven that the matrix M is defined as following:

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

$$A = \begin{bmatrix} \lambda & \frac{-c_x c_y}{\lambda} & \frac{-c_x c_z}{\lambda} \\ 0 & \frac{c_z}{\lambda} & \frac{-c_y}{\lambda} \\ c_x & c_y & c_z \end{bmatrix}, \qquad (20)$$

where $\lambda = \sqrt{c_y^2 + c_z^2}$.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2.3. Camera Calibration Methods

The calibration of perspective camera is analyzed using linear and non-linear methods, as described in the following paragraphs. In the results section, the methods are applied on three projective cameras, in order to estimate the real position of images objects, or construct their volumetric models.

2.3.1. Linear Estimation of the Camera Projection Matrix

The calculation of the extrinsic parameters of the camera, which are the elements of the projection matrix (equivalently camera position, orientation) and the focal length (*f*) is described in this Subsection. If we combine the equations for interior and exterior orientation we obtain an equation that in matrix form can be solved and give the unknown parameters of the calibration. This is the linear estimation of the camera projection matrix.

Let $\mathbf{p}_{s,k} = [x_k, y_k, z_k]^T, k = 1,2,\ldots,N$ be a set of N salient world points (landmarks) and (j_k, i_k) be pixel positions on the image. If we assume the intrinsic parameters known, then their position on the sensor plane (u_k, v_k) can be easily extracted, as described in Subsection 2.2.1. The corresponding sensor coordinates $(u_k, v_k), k = 1,2,\ldots,N$ would be (using Eq.(10)):

$$(u_k, v_k) = \left(\frac{j_k - \frac{N_{col}+1}{2}}{\frac{N_{col}-1}{x_{CCD}}}, \frac{i_k - \frac{N_{line}+1}{2}}{\frac{N_{line}-1}{y_{CCD}}} \right) \quad (21)$$

Given the sensor coordinates, the linear estimation of the camera projection matrix in [2] is accomplished through matrix G, as shown in the following system:

$$G \cdot [R,T] = 0 \Leftrightarrow \begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1z_1 & u_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & v_1x_1 & v_1y_1 & v_1z_1 & v_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2z_2 & u_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & v_2x_2 & v_2x_2 & v_2z_2 & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & u_Nx_N & u_Ny_N & u_Nz_N & u_N \\ 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & v_Nx_N & v_Ny_N & v_Nz_N & v_N \end{pmatrix} \cdot \begin{pmatrix} f \cdot r_{11} \\ f \cdot r_{12} \\ f \cdot r_{13} \\ f \cdot t_x \\ f \cdot r_{21} \\ f \cdot r_{22} \\ f \cdot r_{23} \\ f \cdot t_y \\ r_{31} \\ r_{32} \\ r_{33} \\ t_z \end{pmatrix} = 0 \quad (22)$$

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

However, if we solve Eq.(22) we obtain:

$$u \cdot r_{31} \cdot x_s + u \cdot r_{32} \cdot y_s + u \cdot r_{33} \cdot z_s + t_z - f \cdot s_u \cdot \left(r_{11} \cdot x_s + r_{12} \cdot y_s + r_{13} \cdot z_s + t_x \right) = 0 \quad (23)$$

$$v \cdot r_{31} \cdot x_s + v \cdot r_{32} \cdot y_s + v \cdot r_{33} \cdot z_s + t_z - f \cdot \left(r_{21} \cdot x_s + r_{22} \cdot y_s + r_{23} \cdot z_s + t_y \right) = 0 \quad (24)$$

These equations transformed in matrix form for N salient points $\mathbf{p}_{s,k} = [x_k, y_k, z_k]^T, k = 1,2,\ldots,N$, (using $s_u = 1$), become:

$$\begin{pmatrix} -x_1 & -y_1 & -z_1 & -1 & 0 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1z_1 & u_1 \\ 0 & 0 & 0 & 0 & -x_1 & -y_1 & -z_1 & -1 & v_1x_1 & v_1y_1 & v_1z_1 & v_1 \\ -x_2 & -y_2 & -z_2 & -1 & 0 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2z_2 & u_2 \\ 0 & 0 & 0 & 0 & -x_2 & -y_2 & -z_2 & -1 & v_2x_2 & v_2x_2 & v_2z_2 & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_N & -y_N & -z_N & -1 & 0 & 0 & 0 & 0 & u_Nx_N & u_Ny_N & u_Nz_N & u_N \\ 0 & 0 & 0 & 0 & -x_N & -y_N & -z_N & -1 & v_Nx_N & v_Ny_N & v_Nz_N & v_N \end{pmatrix} \cdot \begin{pmatrix} f \cdot r_{11} \\ f \cdot r_{12} \\ f \cdot r_{13} \\ f \cdot t_x \\ f \cdot r_{21} \\ f \cdot r_{22} \\ f \cdot r_{23} \\ f \cdot t_y \\ r_{31} \\ r_{32} \\ r_{33} \\ t_z \end{pmatrix} = 0 \quad (25)$$

The solution to the system is the eigenvector $\mathbf{g}_{\min}$ of Y, associated with the smallest eigenvalue $\lambda_{\min}$, where:

$$Y = G^T \cdot G \quad (26)$$

The recovery of the exact solution vector will be discussed in the Section of coplanar salient world points.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2.3.1.1. Coplanar Control Points

In the case of N salient coplanar world points $\mathbf{p}_{s,k} = [x_k, y_k, z_k]^T, k = 1,2,\ldots,N$, the above method cannot be used, since $Y = G^T \cdot G$ is singular, $\det(Y) = 0$. If we assume the intrinsic parameters known, (known position on the sensor plane (u_k, v_k), as described in Subsection 2.2.1), the extrinsic camera parameters can be estimated as following.

With no restriction to generality, it can be assumed that $\mathbf{p}_{s,k}$ lie on the XY plane, thus, $z_k = 0$, for all k. In [2], it is proposed to add the right and left sides of equations (23) and (24). The resulting equation is applied for each point and then rewritten in matrix form, describing a homogeneous system of linear equations:

$$\begin{bmatrix} -x_1 & -y_1 & 1 & -x_1 & -y_1 & 1 & (u_1+v_1)\cdot x_1 & (u_1+v_1)\cdot y_1 & (u_1+v_1) \\ -x_2 & -y_2 & 1 & -x_2 & -y_2 & 1 & (u_2+v_2)\cdot x_2 & (u_2+v_2)\cdot y_2 & (u_2+v_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_N & -y_N & 1 & -x_N & -y_N & 1 & (u_1+v_1)\cdot x_N & (u_N+v_N)\cdot y_N & (u_N+v_N) \end{bmatrix} \cdot \begin{pmatrix} f\cdot r_{11} \\ f\cdot r_{12} \\ f\cdot t_x \\ f\cdot r_{21} \\ f\cdot r_{22} \\ f\cdot t_y \\ r_{31} \\ r_{32} \\ t_z \end{pmatrix} = 0 \Leftrightarrow G\cdot P = 0 \quad (27)$$

where $G \in \Re^{N\times 6}$, $P \in \Re^{6\times 1}$.

In [1], equations (23) and (24) are divided and written in matrix form as following:

$$\begin{bmatrix} -v_1\cdot x_1 & -v_1\cdot y_1 & -v_1 & u_1\cdot x_1 & u_1\cdot y_1 & u_1 \\ -v_2\cdot x_2 & -v_2\cdot y_2 & -v_2 & u_2\cdot x_2 & u_2\cdot y_2 & u_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -v_N\cdot x_N & -v_N\cdot y_N & -v_N & u_N\cdot x_N & u_N\cdot y_N & u_N \end{bmatrix} \cdot \begin{pmatrix} f\cdot r_{11} \\ f\cdot r_{12} \\ f\cdot t_x \\ f\cdot r_{21} \\ f\cdot r_{22} \\ f\cdot t_y \end{pmatrix} = 0 \Leftrightarrow G\cdot P = 0 \quad (28)$$

where $G \in \Re^{N\times 6}$, $P \in \Re^{6\times 1}$.

However, in this work, the linear estimation of the camera projection matrix for coplanar objects is performed by keeping both equations for each point $\mathbf{p}_{s,k}$, thus producing a homogeneous linear system of equations, as following:

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

$$\begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & u_1 \cdot x_1 & u_1 \cdot y_1 & u_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & 1 & v_1 \cdot x_1 & v_1 \cdot y_1 & v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_N & -y_N & 1 & 0 & 0 & 0 & u_N \cdot x_N & u_N \cdot y_N & u_N \\ 0 & 0 & 0 & -x_N & -y_N & 1 & v_N \cdot x_N & v_N \cdot y_N & v_N \end{bmatrix} \cdot \begin{pmatrix} f \cdot r_{11} \\ f \cdot r_{12} \\ f \cdot t_x \\ f \cdot r_{21} \\ f \cdot r_{22} \\ f \cdot t_y \\ r_{31} \\ r_{32} \\ t_z \end{pmatrix} = 0 \Leftrightarrow G \cdot P = 0 \quad (29)$$

where $G \in \Re^{N \times 9}$, $P \in \Re^{9 \times 1}$.

By solving the above homogeneous system of equations using Eq.(26), the unknown quantities in P are estimated using $\mathbf{g}_{\min} = [g_1 \quad g_2 \quad \cdots \quad g_9]^T$ with an unknown multiplying factor s, since $\|\mathbf{g}_{\min}\| = 1$. Thus the first 2 columns of the rotation matrix can be estimated.

$$\begin{pmatrix} s \cdot f \cdot r_{11} & s \cdot f \cdot r_{12} & ? \\ s \cdot f \cdot r_{21} & s \cdot f \cdot r_{22} & ? \\ s \cdot r_{31} & s \cdot r_{32} & ? \end{pmatrix} \quad (30)$$

Equivalently, the elements of the full rigid transformation in homogeneous coordinates that can be estimated directly by $\mathbf{g}_{\min}$ are shown below:

$$\begin{pmatrix} s \cdot f \cdot r_{11} & s \cdot f \cdot r_{12} & ? & s \cdot t_x \\ s \cdot f \cdot r_{21} & s \cdot f \cdot r_{22} & ? & s \cdot t_y \\ s \cdot r_{31} & s \cdot r_{32} & ? & s \cdot t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & ? & g_3 \\ g_4 & g_5 & ? & g_6 \\ g_7 & g_8 & ? & g_9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (31)$$

Factor s is an arbitrary coefficient without physical meaning, induced by homogeneity of the linear system of equations, whereas factor f is the focal length of the camera that is considered unknown, even when the affine transformation between (j,i) and (u,v) is known. Recovering f and s is described below.

Recovering focal length f through exhaustive search

As it can be seen in Eq.(6) and equations Eq.(23), Eq.(24), since in this work we assume that $s_u = 1$, the coefficients of x_s and y_s are multiplied by the focal length (f). Since $[r_{11}, r_{21}, r_{31}]^T, [r_{12}, r_{22}, r_{32}]^T$ are by definition orthonormal vectors, it is proposed to exhaustively search the focal length by considering the two vectors **w** and **q** as following:

$$\mathbf{w} = \left[\frac{g_1}{f}, \frac{g_4}{f}, g_7, 0\right]^T$$
$$\mathbf{q} = \left[\frac{g_2}{f}, \frac{g_5}{f}, g_8, 0\right]^T \quad (32)$$

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

and testing their orthogonality. Specifically, the value of *f* is obtained by requiring $\frac{\mathbf{w}\cdot\mathbf{q}}{|\mathbf{w}||\mathbf{q}|}=0$. An example of focal length recovery through the aforementioned procedure is shown in Fig.6.

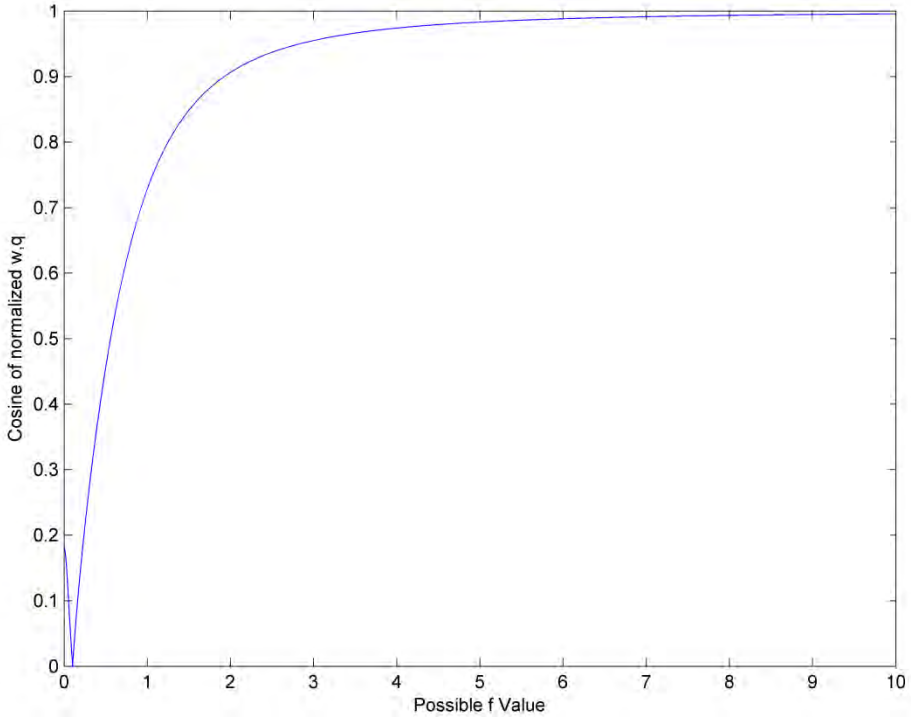


Figure 6. The result of searching focal length (*f*) through exhaustive search, in the case of a random camera optical axis. As it can be observed through the orthogonality validity, it is assumed that *f*=0.1.

Handling special cases: camera vector parallel to X, Y and Z scene axis

Orthogonality of *w*, *q* is impossible to test when the vector of the camera optical axis **v**, is [0,1,0] or [1,0,0] and the inner product would always conclude to zero.

In this case, the second consequence of the orthonormality property of the Rotation Matrix may be used, namely the equal length of the vectors of two first columns of the matrix. Thus, the value of *f* can be obtained by requiring $\|w\|=\|q\|$ (Fig.7).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

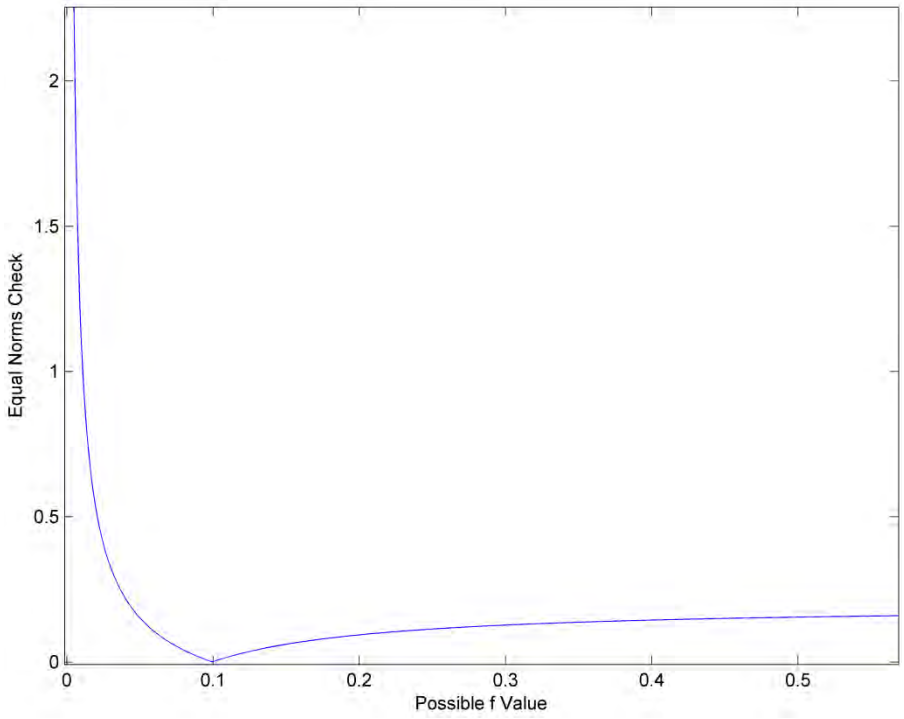


Figure 7. The result of searching focal length (*f*) through exhaustive search, in the case of a camera optical axis parallel to *X* axis (assumed *f*=0.1).

In the case of camera being parallel to Z real world axis of the (Z_s in Fig.3) focal length cannot be computed, because the axis Z_s is perpendicular to image plane.

Recovering the unknown factor s and estimating the exact solution of the transformation matrix

If the value of the focal length has been estimated, then we can decouple it from the matrix in Eq.(31) and update it as following:

$$\begin{pmatrix} \frac{s\cdot f\cdot r_{11}}{f} & \frac{s\cdot f\cdot r_{12}}{f} & ? & s\cdot t_x \\ \frac{s\cdot f\cdot r_{21}}{f} & \frac{s\cdot f\cdot r_{22}}{f} & ? & s\cdot t_y \\ s\cdot r_{31} & s\cdot r_{32} & ? & s\cdot t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s\cdot r_{11} & s\cdot r_{12} & ? & s\cdot t_x \\ s\cdot r_{21} & s\cdot r_{22} & ? & s\cdot t_y \\ s\cdot r_{31} & s\cdot r_{32} & ? & s\cdot t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} g_1' & g_2' & ? & g_3 \\ g_4' & g_5' & ? & g_6 \\ g_7 & g_8 & ? & g_9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (33)$$

The resulting vector $\mathbf{g}_{\min} = \begin{bmatrix} g_1' & g_2' & g_3 & g_4' & g_5' & g_6 & g_7 & g_8 & g_9 \end{bmatrix}^T$ (after the extraction of f from the corresponding elements) is multiple of the real solution with an unknown multiplying factor (s):

$$sol = s\cdot\mathbf{g}_{\min} \quad (34)$$

This factor (34) may be found by noting that the rotation matrix is orthonormal and concluding to a quadratic form [1]:

$$s^4 - s^2\left(r_{11}^2 + r_{12}^2 + r_{21}^2 + r_{22}^2\right) + \left(r_{11}r_{22} - r_{12}r_{21}\right)^2 = 0 \quad (35)$$

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Thus, factor s is found through the roots of Eq.(35). Then, the first two rows of the rotation matrix are divided by s. Finally, the third row of the rotation matrix is found by the cross-product of the first two rows.

However, in this work it is proposed to use normalization (divide each vector by its length) in order for the factor to be ignored as following:

$$\mathbf{w}' = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$
$$\mathbf{q}' = \frac{\mathbf{q}}{\|\mathbf{q}\|} \quad (36)$$

As far as the vector that holds the translation data is concerned (namely the $\mathbf{e} = [g_3, g_6, g_9, 1]^T$), it may be divided by any of the two aforementioned norms since $\|w\| \cong \|q\|$. Specifically, the referential vector will become:

$$\mathbf{e}' = \frac{\mathbf{e}}{\|\mathbf{w}\|} \Leftrightarrow \mathbf{e}' = \frac{\mathbf{e}}{\|\mathbf{q}\|} \quad (37)$$

As far as the sign of the three vectors ($\mathbf{w}', \mathbf{q}'$ *and* $\mathbf{e}'$) is concerned, the translation at *Z* axis is an auxiliary landmark, since it is known that it should not surpass the level of the ceiling. In real world (scene) of the experimental setup of this work the level of the ceiling is set to zero and *Z* axis is positive towards the floor of the room, as shown in Fig.8. Therefore, t_z should be a positive number, since the camera vector $\mathbf{c} = (c_x, c_y, c_z)$ is equal to $(-t_x, -t_y, -t_z)$, where $[t_x, t_y, t_z, 1]^T$ is the last column of the translation matrix *T* of Eq.(2).

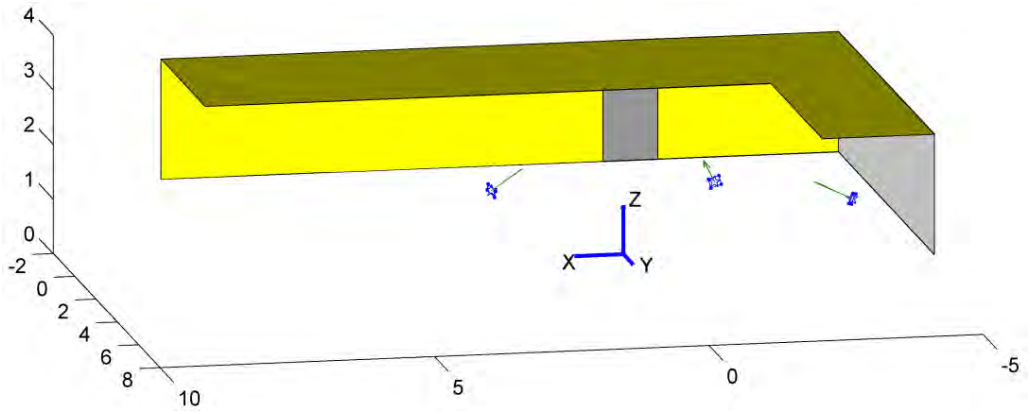


Figure 8. Simulation of real room experimental setup of this work, along with its coordinate system and the positions of the three projective cameras calibrated and used in the applications of calibration.

After the application of Eq.(36) and (37), the elements of the full rigid transformation of Eq.(33) are updated as following:

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

$$\begin{pmatrix} r_{11} & r_{12} & ? & t_x \\ r_{21} & r_{22} & ? & t_y \\ r_{31} & r_{32} & ? & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} g_1'' & g_2'' & ? & g_3' \\ g_4'' & g_5'' & ? & g_6' \\ g_7' & g_8' & ? & g_9' \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left[\mathbf{w}', \mathbf{q}', ?, \mathbf{e}'\right] \quad (38)$$

The third row of the rotation matrix (**d**) may be found by exploiting the property of orthogonality. Thus, the computation of the cross-product of the first two rows of the full rigid transformation matrix concludes exactly to the third column we are looking for:

$$\mathbf{d} = \mathbf{w}' \times \mathbf{q}' \quad (39)$$

Finally, the exact solution vector is:

$$\mathbf{g}_{\min}' = \left[g_1'' \quad g_2' \quad g_3' \quad g_4'' \quad g_5' \quad g_6' \quad g_7' \quad g_8' \quad g_9'\right]^T \quad (40)$$

and the full rigid transformation matrix is:

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left[\mathbf{w}', \mathbf{q}', \mathbf{d}, \mathbf{e}'\right] \quad (41)$$

Therefore, the rotation *R* and translation *T* matrices of Eq.(11) are finally known.

Recovering the camera position and camera axis

The position of each camera in the scene coordinate system $\{(x_{cam}, y_{cam}, z_{cam})_c\}$ is given by $T_{cam} = R^T\left[-t_x, -t_y, -t_z\right]^T$, where $R^T = R^{-1}$ is the inverse rotation matrix (R) – of Eq.(1). Moreover, the optical axis of the camera is obtained by R^{-1} multiplied by the camera *Z* axis, as following:

$$\mathbf{v} = R^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (42)$$

The *Z* axis rotation matrix of Euler angles (R_z) of the Eq.(19), is obtained by the inverse matrix of rotation of the vector $\mathbf{v}$ of the camera optical axis with the *Z* axis multiplied by the rotation matrix R of the exact solution vector (Subsection 2.3.1.1-3). The angle of the vector of the camera optical axis is obtained by the inverse cosinus of the first element of the *Z* axis rotation matrix of Euler angles (R_z).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2.3.2. *Nonlinear Calibration Method*

The Tsai camera model [12] was partially utilized, while using the known intrinsic parameters of the cameras (sensor size and focal length). The definition of the extrinsic parameters was slightly modified, as following: since the cameras were firmly installed on the Laboratory's walls, their positions $\{(x_{cam}, y_{cam}, z_{cam})_c\}$ were measured and used for subsequent calculations (c=1, 2, 3 denotes the available projective cameras). The orientation of the cameras was defined by the directions of view $\mathbf{N}_c$ and the rotation of the sensor round $\mathbf{N}_c$ by angle θ_c.

The extrinsic parameters ($\mathbf{N}_c$, θ_c) were derived using the following process. A set of N_p=24 points were marked on the floor (Fig.9) and their real world coordinates $\{(x^q_{real}, y^q_{real}, z^q_{real})\}$, ($q$=1,2, …,$N_p$) as well as their position in the frame of each projective camera c $\{(c_q, r_q)_c\}$ were manually measured. Moreover, the frame of reference was transformed through translation and rotation, where the sequence was translation followed by rotation, instead of rotation followed by translation that was applied in the proposed linear method. Thus, the matrix form of the multiplication of the set of the N_p points was $R \cdot T$, where R as in Eq.(1), however camera coordinates are contained directly in T, since $T = \begin{pmatrix} 1 & 0 & 0 & -x_{cam} \\ 0 & 1 & 0 & -y_{cam} \\ 0 & 0 & 1 & -z_{cam} \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

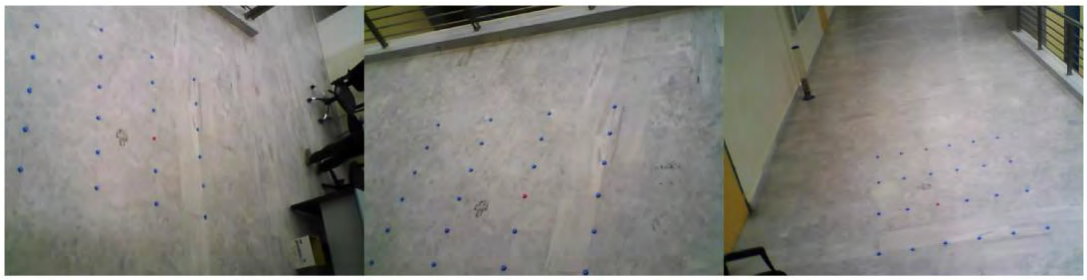


Figure 9. Frames from the tree projective cameras, all including the 24 marked on the floor points.

Furthermore, the functions of projecting the points on the camera system (by using the projection matrix of Tsai [12]) and applying the affine transformations were applied for the three cameras. Thus, the expected frame positions $\{(Ec_q, Er_q)_c\}$ were calculated for each camera c.

The extrinsic parameters for each camera were obtained by using the Matlab implementation of the Nelder-Mead Simplex Method [13], by minimizing the mean error $error = \frac{1}{N_p} \sum_{q=1}^{N_p} \left\| (c_q, r_q)_c - (Ec_q, Er_q)_c \right\|$. The error is modified to penalize points imaged outside the CCD (image sensor) of each camera.

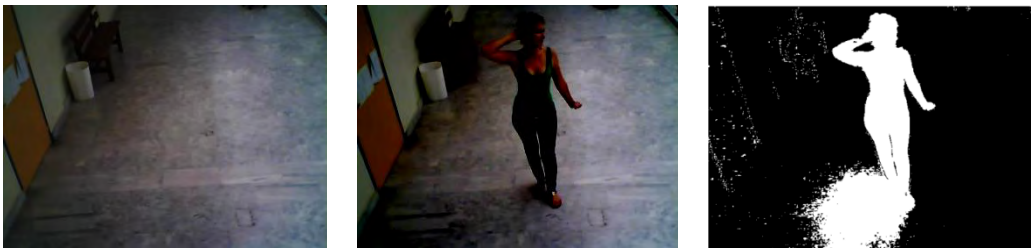
Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

2.4. Computer Vision Applications

After the calibration of the three identical projective cameras is applied, frames containing an object/person are acquired by them. The frames get segmented by a background subtraction segmentation method. The position of the object may be estimated in real time through the segmented frames and based on the calibration of each camera. Moreover, the digital shape of the object can be generated by having the segmented frames that contain the binary silhouette and the calibration of each camera and applying an S2S algorithm.

2.4.1. *Segmentation method - Foreground detection*

A simple foreground segmentation algorithm is used, which detects foreground objects based on changes between the current and a reference frame. Background images (Fig.10,a) and images containing the object of interest (Fig.10,b) are acquired within few minutes, to exclude variations of lighting conditions. Thresholding is applied on the grayscale images derived from background subtraction, in order to create the segmented images. A typical original and the corresponding segmented frame containing the silhouette, along with noisy white pixels and shadow, are shown in Fig.10, (b) and (c), respectively. In general, noise and shadows confound scene interpretation. However, the proposed S2S algorithm remains unaffected.



(a) **(b)** **(c)**

Figure 10. Two typical original frames acquired by one of the three projective cameras: a background image (a) and an image containing the object/human model (b). The corresponding segmented frame is shown in (c).

2.4.2. *Position Estimation from mono-ocular video*

The real world position of an object, also in the case of a human subject, may be estimated from a single camera view (mono-ocular) under the following conditions:

- The intrinsic and extrinsic parameters of the cameras are known, through the calibration
- The segmentation of the silhouette is accurate
- The object touches the ground and the area of its base is small
- The shape of the ground surface is known.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

More specifically, let the non-zero pixels of the segmented silhouette, of each camera $c=1,2,3$ frame, be $\left(j_p, i_p\right)_c$, $p=1,2,\ldots,M_c$, where M_c is the number of segmented silhouette pixels. For any pixel of the silhouette, we can obtain its position, as if the pixel were lying on the floor (assuming known room floor), as following.

The corresponding sensor coordinates $\left\{\left(u_p, v_p\right)_c\right\}$ ($c=1,2,3$ denotes the available projective cameras) are easily obtained from the equations of the affine transformations Eq.(7), as following:

$$\begin{aligned} u_p &= \frac{j_p - c_1}{a} \\ v_p &= \frac{i_p - c_2}{b} \end{aligned} \tag{43}$$

Then, real world coordinates of each point $\mathbf{p}_s = (x_s, y_s, z_s)$ of the model/object that can be viewed by each camera would be found through the sensor coordinates (u_p, v_p) of each camera c.

Let $\mathbf{k}_c$ be the vectors of each camera that points to (u_p, v_p) of each camera – $\left\{\left(u_p, v_p\right)_c\right\}$. For a sample camera c, vector $\mathbf{k}$ will be the following:

$$\mathbf{k} = \left(\frac{u_p}{f}, \frac{v_p}{f}, 1\right) \tag{44}$$

where f is the principle distance of the sample camera, as analyzed in Subsection 2.3.1.1. The coordinates of $\mathbf{k}_c$ in scene (real world) system of reference will be:

$$\mathbf{b}_c = \left(b_1, b_2, b_3\right) = R_c^{\,T} \cdot \mathbf{k}_c \tag{45}$$

where $R_c^{\,T}$ is the transpose of the rotation matrix of Eq.(1) for each camera c, found through the calibration (Eq.(41)) of each camera. Finally, for a sample camera, the real world points $\mathbf{p}_s = (x_s, y_s, z_s)$ that are imaged on the same pixel are given by the following relations:

$$\begin{aligned} x_s &= x_0 + b_1\tau \\ y_s &= y_0 + b_2\tau \\ z_s &= z_0 + b_3\tau \end{aligned} \tag{46}$$

where (x_0, y_0, z_0) is the camera position and $\tau>0$ the parameter of the line that emanates from the camera center of projection and passes through the corresponding pixel.

In the frame of reference of the real world (scene) of the experimental setup of this work, the plane of the ceiling is set to zero and Z axis is positive towards the floor,

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

as shown in Fig.8. Therefore, parameter τ, for b_3 and the real world coordinates of the position on the floor, becomes:

$$\tau = \frac{z_{\max} - z_0}{b_3} \quad (47)$$

If τ is replaced in Eq.(46), then real world coordinates of the equation correspond to the real world coordinates of the position of the human model or object on the floor, as it is viewed by the sample camera.

The real position of the silhouette is approximated more closely, if we select the image pixel with the maximum of the Z axis ($z_{\max}$) coordinate of $\mathbf{b}_c$, since the coordinates of $\mathbf{b}_c$ that correspond to $z_{\max}$ resembles the vectors that points to the lower point of the segmented silhouette, which is identical to its position on the floor.

The same procedure would be followed for each available camera, in order to find the real world coordinates of the position of the model on the floor as it is viewed by each one of the available cameras. Finally, the average of the resulting real world coordinates for the view and data of each one of the available cameras would conclude to three coordinates that will approach the exact real position of the model while the number of the cameras increases.

2.4.3. *Silhouette to Shape Algorithm*

The first requirement of Silhouette to shape method (S2S, commonly referred as shape from silhouette -SfS), in order to reconstruct the object of interest, is the set of frames, acquired by one or more cameras, that contain the object. Then, the silhouette of these images must be segmented. The algorithm used in this work is the space carving algorithm.

The methodology of S2S would be easily explained through a graphical 2D example: Let the red rectangle of Fig.11 be the object of interest that will be reconstructed and that lies in the blue VOI. Four cameras of different extrinsic and intrinsic parameters can view the object, as shown in Fig.11. The intrinsic parameters of camera 1 (the image sensor in gray line, the focal spot in double arrow with dashed line, the center of projection as a blue star and the vector of the camera optical axis as a green arrow), as well as the position of the projection of the object on the image sensor of the camera are also depicted. Furthermore, it is shown that any real world point along each line (yellow circles on a sample line), which focuses on the center of projection, corresponds to the same position (yellow circle on magenta line of projected object) on image sensor.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

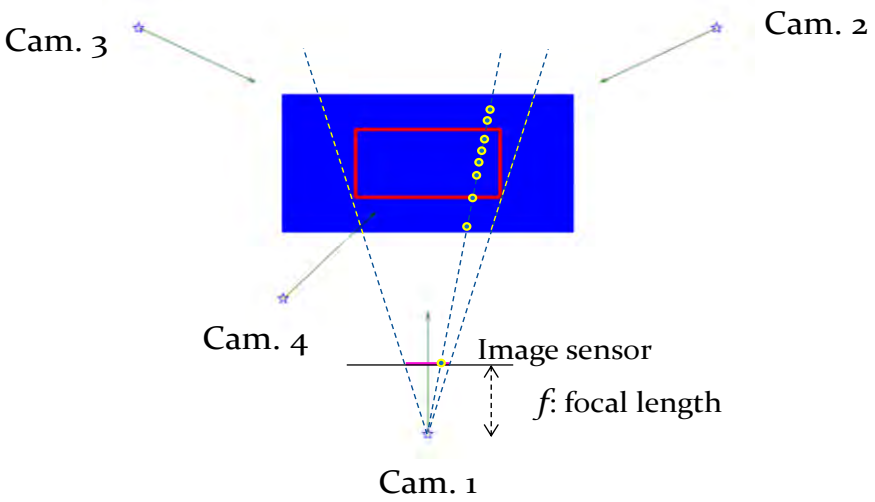


Figure 11. A graphic representation of S2S algorithm in 2D. The four cameras and their optical axis are shown in blue stars and green arrows. The object of interest is shown in red color and the VOI in blue. Concerning camera 1, the image sensor (gray line), the focal spot (double arrow with dashed line), the center of projection (blue star) and the vector of the camera optical axis (green arrow) are depicted. The projection of the object on the image sensor is shown in magenta line. The yellow circle on the magenta line indicates that any real world point (yellow circles) along this line corresponds to the same position of CCD.

The generalized cone of the view of camera 1 is shown in green on the left diagram of Fig.12, superimposed with the VOI and the real 2D object to show that the 2D object lies inside the volume generated by back-projecting the silhouette area of only one camera view. The intersection of the generalized cone with the VOI generates the initial optic hull of the object. This limiting smallest volume is shown from an upper view, as a 2D polyhedron in blue lines, at the right side of Fig.12.

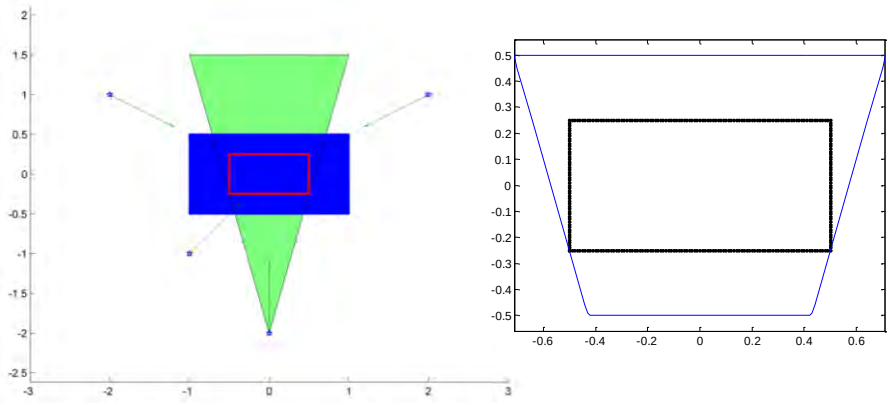


Figure 12. On the left diagram, the four cameras (blue stars) of the setup of the example and the vectors of each one's optical axis (green arrows) are depicted. Moreover, the generalized cone of the view of camera 1 (green) is shown superimposed with the VOI (blue) and the real 2D object (red). On the right diagram, the visual hull, as a 2D polyhedron (blue lines), and the real object (black rectangle) are shown from an upper 2D view.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

The generalized cone of the view of camera 2 is shown in green on the left plot of Fig.13, superimposed with the VOI and the real 2D object to show that the 2D object lies inside the volume generated by back-projecting the silhouette area of each camera view. The intersection of the generalized cone of this camera with the one of camera 1 and the VOI generates a new optic hull of the object. This optic hull is shown from an upper view, as a 2D polyhedron in blue lines, at the right side of Fig.13.

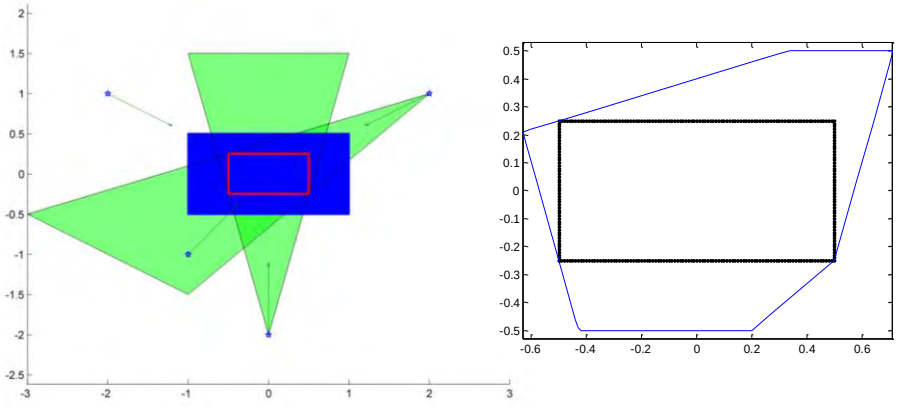


Figure 13. Reconstructing shape from 2 views. On the left diagram, the new generalized cone (green) of the view of camera 2 appears. On the right diagram, the new visual hull of the intersection of the two cones with the VOI is shown as a 2D polyhedron (blue lines) from an upper 2D view.

The same steps are followed for the other two views of the other two cameras, as shown in Fig.14 and Fig.15. The final reconstructed volume from the intersections of the generalized cones of four cameras views is shown on the right of Fig.15.

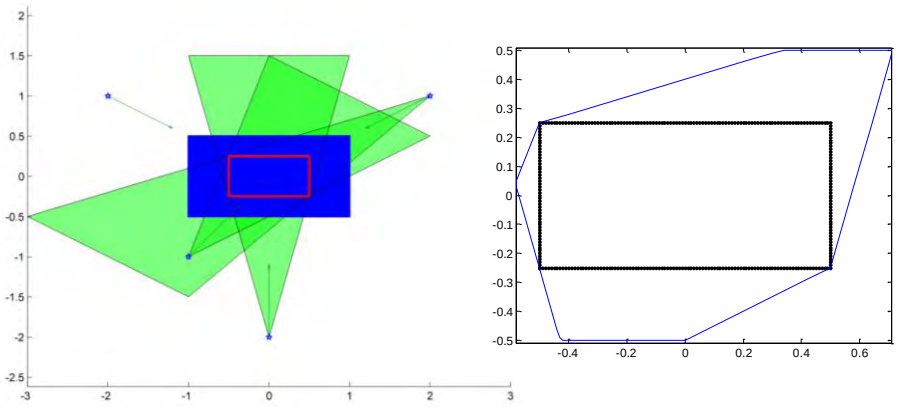


Figure 14. Reconstructing shape from 3 views. On the left diagram, the new generalized cone (green) of the view of camera 3 appears. On the right diagram, the new visual hull of the intersection of the 3 cones with the VOI is shown as a 2D polyhedron (blue lines) from an upper 2D view.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

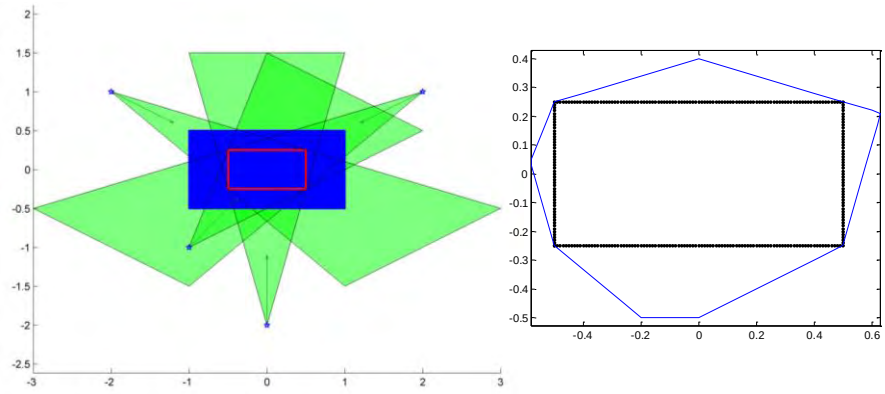


Figure 15. Reconstructing shape from 4 views. On the left diagram, the new generalized cone (green) of the view of camera 4 appears. The final reconstructed volume from the 4 cameras view is shown on the right as a 2D polyhedron (blue lines) from a 2D upper view.

The steps of the S2S algorithm in this work are the following. As a first generic step, the generalized cone of each camera that contains the imaged object in 3D space is generated using the corresponding images**.** A volume of interest (VOI) that includes the object in 3D space is considered (*V*). VOI is tessellated into elementary volumes $\delta V=\delta l \times \delta l \times \delta l$. Each δV is projected on the frame acquired by each camera, by using the calibration of each camera. Finally, the volumetric model is constructed through the δVs that are considered to belong to the object.

A δV belongs to the object when its projection lies on white pixels on all silhouette-frames. The set of δVs that project in non-zeros pixels in all available silhouette frames construct the smallest volume obtainable that contains the object in 3D. They are the discrete intersections of the optic hulls from each camera, so they construct a volume that approaches the 3D representation of the object. These steps are figured in the block diagram of Fig.16.

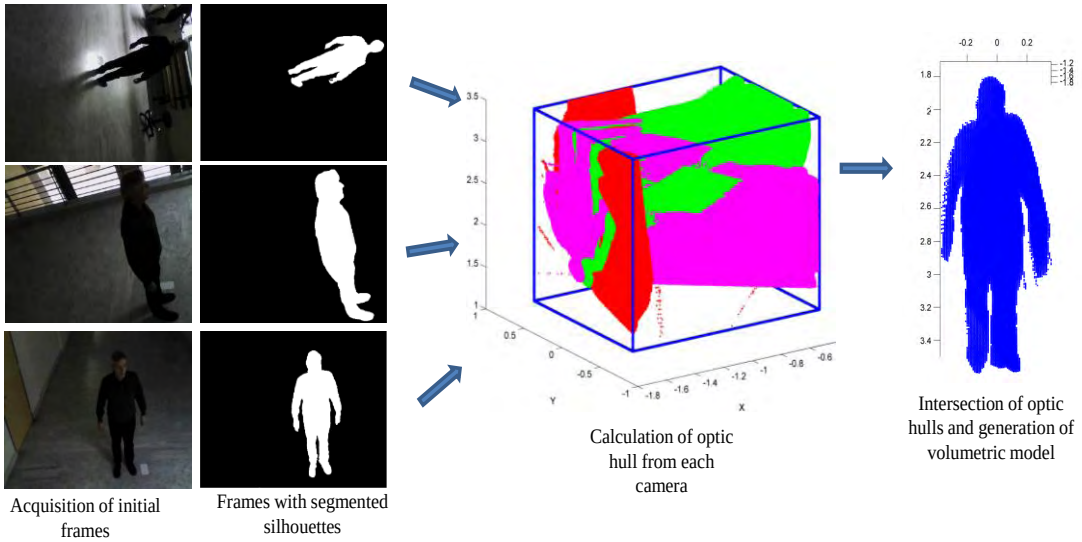


Figure 16. The steps of the S2S algorithm in a real data example.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

For the proposed application, let B_c be the binary frame containing the silhouette imaged by camera *c* and F_c the calibration function that maps a real world point $\mathbf{x}_{\mathbf{real}} = (x_{real}, y_{real}, z_{real})$ to image coordinates of frame B_c. Furthermore, a cube *C*, large enough to include the common field of view (FoV), is initialized as the VOI (Fig.17) and divided into elementary volumes δV of dimensions $\delta l \times \delta l \times \delta l$. Typical value of δl is 1 cm, although this depends on memory, as well as accuracy requirements. The employed silhouette to shape algorithm returns a binary volume *V*, which is the discretized cube *C* and contains the optic hull.

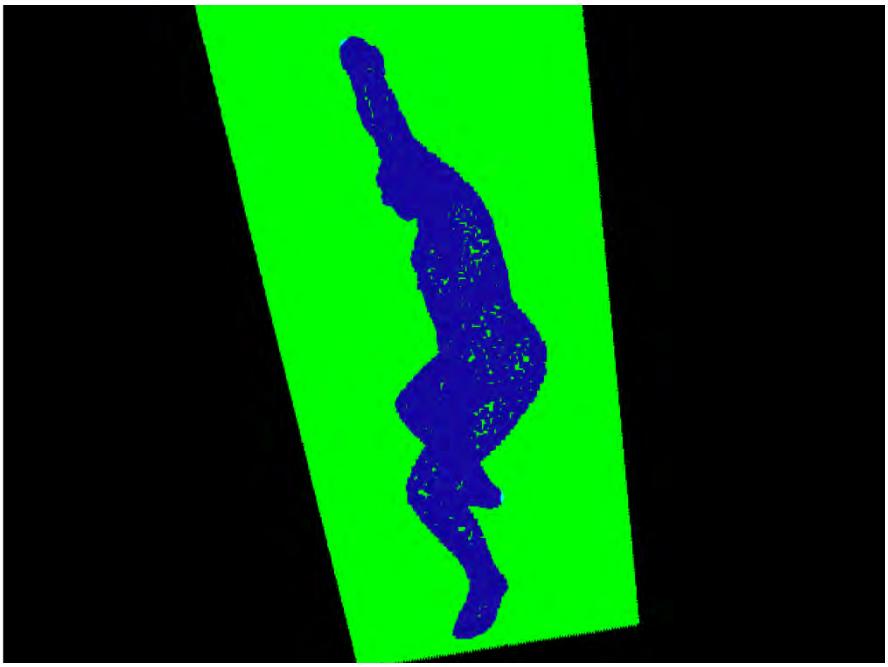


Figure 17. Rendered synthetic model frame of a projective camera. The cube *C* used to produce the binary volume *V* is indicated in green, whereas the synthetic 3D model is also rendered.

The employed S2S algorithm is described in the following pseudocode:

```
Initialize VOI as the cube C, enclosing the object
FOR each projective camera c
 FOR each δV in C
  IF B_c(F_c(δV)) = 1, for c=1,2,3
  THEN
      Obtain indices (i,j,k) of δV with respect to VOI
      Set V^c_{i,j,k} = 1
  END
 END
 IF c==1 V^c = V^c ∩ V^f ELSE V^c = V^c ∩ V^{c-1}
END
```


Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

3. Results

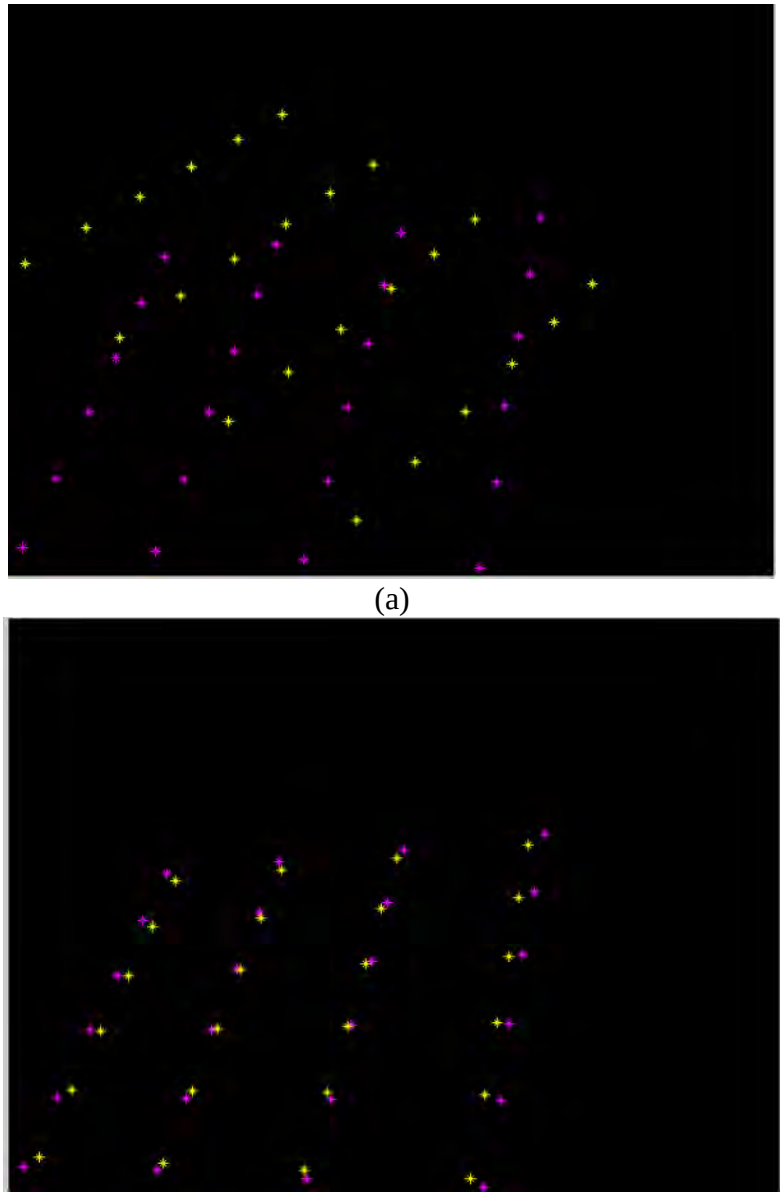
Quantitative and qualitative results for camera calibration methods, S2S and position estimation method using three projective cameras are presented in this section, using both synthetic and real data. Three projective cameras (D-Link DCS-930/L) were used, placed at known positions in the room in order to share a common portion of real world space as their FoV. A virtual representation of the real room is presented in Fig.8. The pixilation of each frame is 480x640.

3.1. Calibration Results

The nonlinear calibration method, as described in Subsection 2.3.2, was applied to all 3 cameras. The positions ($\{(x_0, y_0, z_0)_c\}$, $c=1,2,3$) and the intrinsic parameters (sensor size and focal length) of all 3 cameras were known. A number of N_p=24 points were marked on the floor and their real world coordinates were manually measured. The Matlab programming environment was used for the implementation of the described algorithms, with CCD dimensions restrictions on projecting points, and outputted the extrinsic parameters (optical axis $\mathbf{N}_c$ and angle θ_c round $\mathbf{N}_c$) for each camera c.

In Fig.18(a), the initial step of the optimization algorithm is shown for one of the three cameras. The ground truth of the N_p=24 points on the floor, as the camera views them, is depicted as magenta stars. The re-projected points of the initialized parameters that the optimization algorithm should match with the ground truth points are depicted as yellow stars. In the beginning, the parameters of the initialization differentiate from the real extrinsic parameters of the camera. Consequently, the initial synthetic points are being re-projected far from the points of the ground truth (Fig.18,a). However, after several steps of the optimization algorithm, with the aforementioned known parameters and restrictions, the values of the resulting extrinsic parameters are very close to the real extrinsic parameters of the camera. Thus, the points that are re-projected with the resulting parameters are approximately identical to the real world points on the floor, as shown in Fig.18(b).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7



(a)

(b)

Figure 18. The initial and final step of the Nelder-Mead Simplex Method for the non-linear calibration method. The real points are shown in magenta dots and the re-projected points, using the calibration data, are shown in yellow dots.

The results of the nonlinear calibration method for all the three projective cameras are shown through the re-projection of the points, by using the calibration data of each camera, as magenta stars, along with the real 24 points marked on the floor (blue dots), on the frames of real world of the room (Fig.19) where the projective cameras are mounted.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Figure 19. Frames from the tree projective cameras, all including the 24 marked on the floor points (blue dots) superimposed by the results (magenta stars) of the non-linear calibration method.

Results of both linear and nonlinear method are shown on the diagram of Fig.20 that resembles the charged-coupled device image sensor of one of the three cameras.

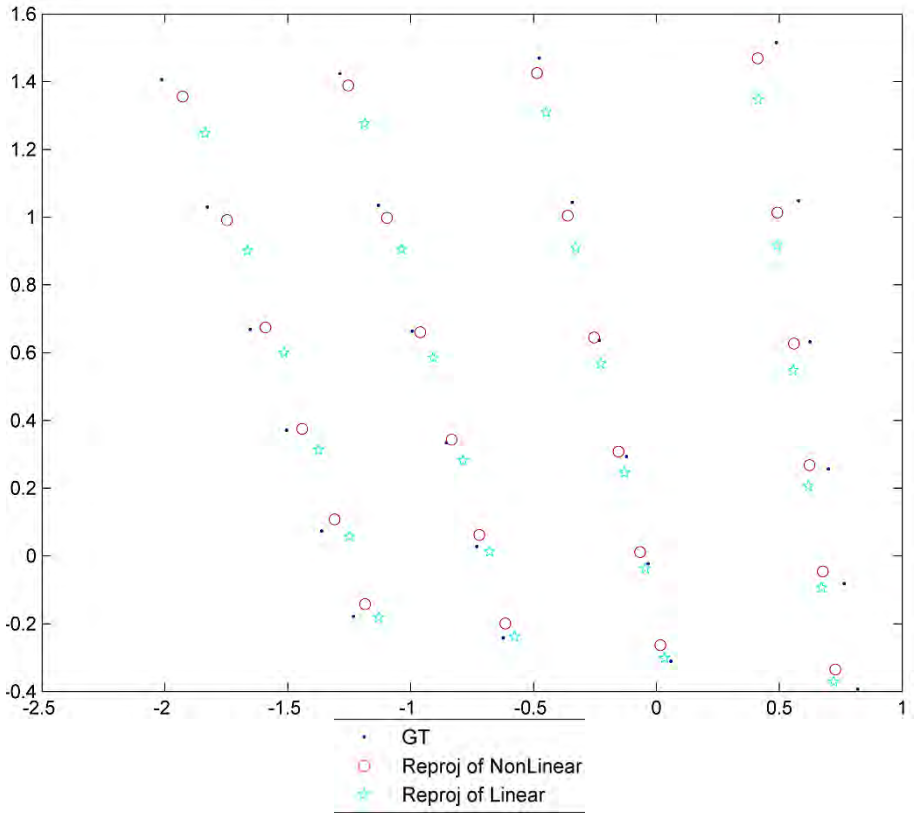


Figure 20. Results of both linear and nonlinear calibration method for one of the three projective cameras plotted on its image sensor. For the set of 24 caps (ground truth positions – GT) are shown in blue dots, the re-projection of the results of the nonlinear and of the linear method are shown in red cycles and cyan stars, respectively.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

3.2. *Results for Computer Vision Applications*

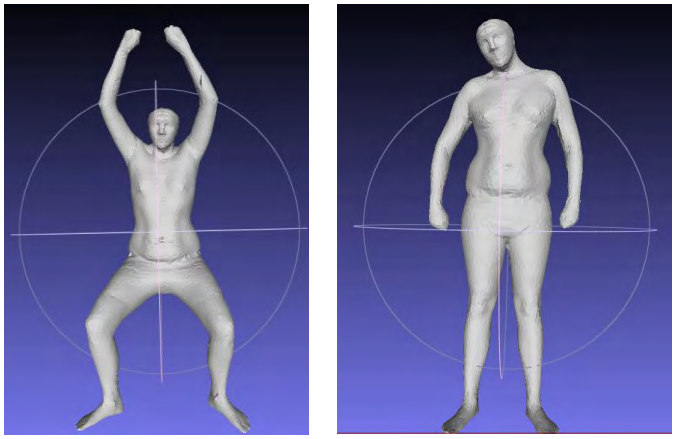
Results of the computer vision applications – of position estimation and S2S methods – are both quantitative and qualitative, since they are presented using both synthetic and real data for both applications.

3.2.1. *Real-Position Estimation*

Both synthetic and real models were rendered or captured, respectively, by the three projective cameras (the placement of which was specific in order to share common portion of real world space as their FoV). The exact position of each camera is shown in a virtual representation of the real room in Fig.8. The cameras' calibration was used in order to estimate the model's position.

3.2.1.1. Synthetic Data

Synthetic data are used in this work to quantify the error of the position estimation method, since the exact position of the model is predefined by the user and the process is not affected by segmentation and calibration. Two 3D models (Fig.21) obtained from [14], [15] were utilized. These models are in the form of triangulated surfaces. However, only the coordinates of the vertices are used for rendering the binary silhouette frames



(a) (b)

Figure 21. The 3D human models used this work: model *i* (a) and model *ii* (b).

The 3D human models, with known real world points, are virtually placed in room at a certain position within the FoV of all the projective cameras (Fig.22).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

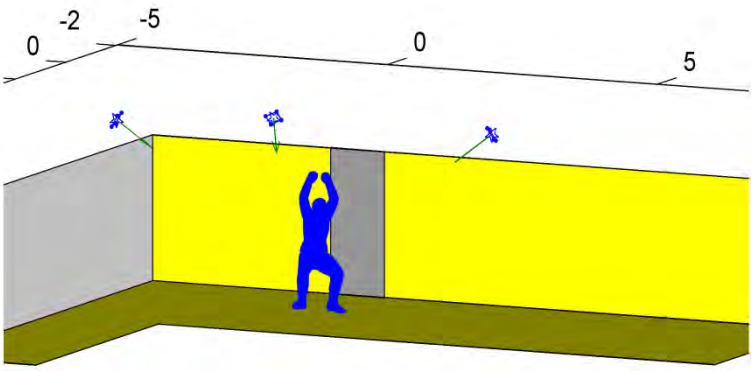


Figure 22. 3D representation of the room with the three projective cameras (with orientation indicated by normal vectors) and the synthetic 3D human model *i* placed at a certain-known position in the room.

Each real world point (vertex) of the 3D human model is rendered according to calibration. More specifically, the corresponding binary frames B_c (Fig.23,a-c), that contain the silhouette are generated using functions F_c (see Subsection 2.4.3).

The model of Fig. 21(a) is placed at a certain position with real world coordinates: $(x\_real, y\_real) = (-0.4, -0.1)$. The proposed position estimation method of Subsection 2.4.2 is applied and the estimated value of the position (real world coordinates) is re-projected on each segmented frame as a red-filled star, as shown in Fig.23, in order to be visually tested. The real world coordinates of the position of the synthetic human model on the floor were estimated using the view of each one of the available cameras. Each position estimation value is indicated at the corresponding binary frame (Fig.23,a-c). The average of the estimated real world positions of the available cameras approaches the exact real position of the model – the *Z* axis coordinate has always the value on which the floor is. The final position is plotted as a red star in a sample frame B_c in Fig.23(d).

(a) (b)

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

(c) (d)

Figure 23. Position estimation result for synthetic data of model *i* plotted as red-filled stars on the corresponding rendered frame of each camera (a-c). The average of the three different position estimations indicated in randomly one of the three rendered frames.

The clipping of the model that is observed at the upper part of the hands in two of the projective silhouettes is due to the placement of the model at a position to which the entire model is not within the FoV of all the projective cameras. However, this issue does not affect the algorithm of position estimation.

Moreover, the final estimated position of another model placed in the same position as the previous model is also indicated through a red star in Fig.24.

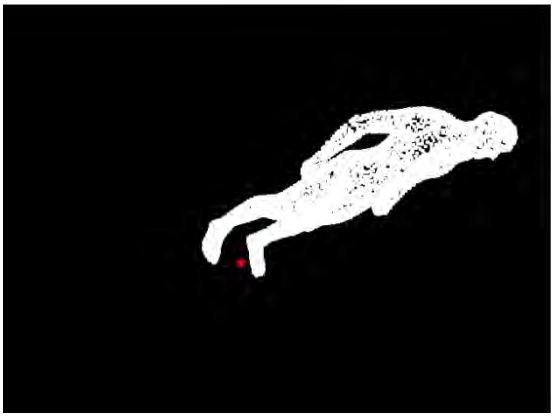


Figure 24. Final position estimation result for synthetic model *ii* plotted as red-filled stars on a randomly selected frame of the three rendered frames.

Both models were also placed in a different position $(x\_real, y\_real) = (0,0)$ and their position was estimated again. In Fig. 25(a-c), each position estimation value is indicated at the corresponding binary frame for model of Fig.21(b). The final position is plotted as a red star in a sample frame B_c in Fig.25(d). The final estimated position of the model of Fig.21(a), which is also placed in this position, is indicated by a red star in Fig.26.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

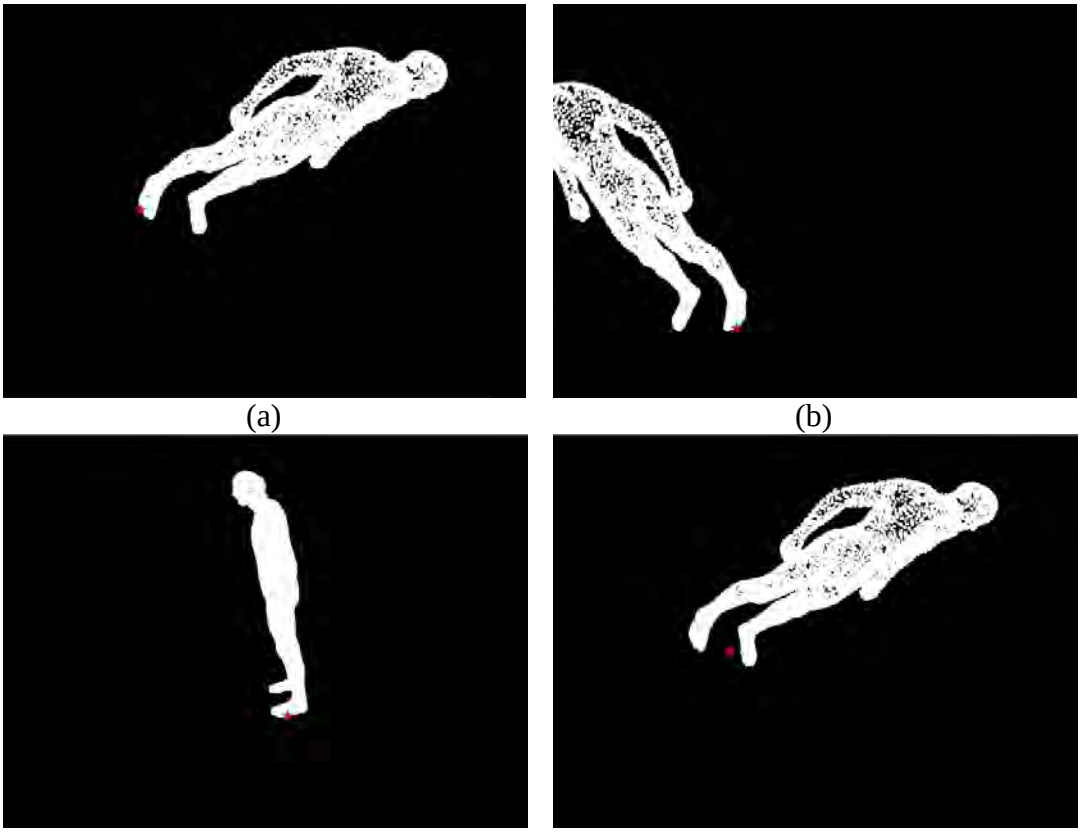


Figure 25. Position estimation result for synthetic model *ii* plotted as red-filled stars on the corresponding rendered frame of each camera (a-c). The average of the three different position estimations indicated in randomly one of the three rendered frames.

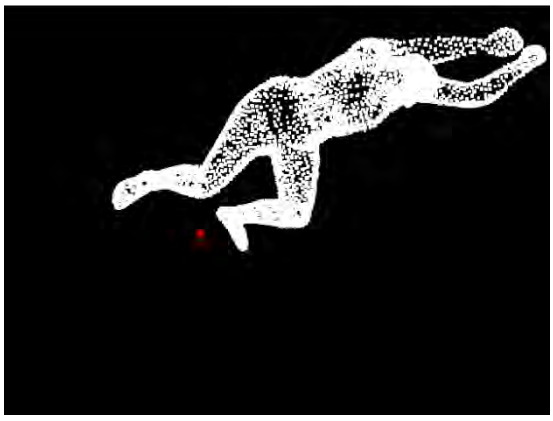


Figure 26. Final position estimation result for synthetic data of model *i* plotted as red-filled stars on a randomly selected frame of the three rendered frames.

Furthermore, numeric results with the values of the positions on *X* and *Y* axis (the *Z* axis coordinate has always the value on which the floor is) are indicated in Table I for the two determined positions (x_real, y_real) of the synthetic human models and the estimated positions (x_est, y_est) by the proposed method of position estimation.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Table I

Comparison of the determined position and the estimated by the proposed method of position estimation for the two synthetic human models.

	Model 1	**Model 2**
Position 1: (-0.4,-0.1)	(-0.5530, -0.2171)	(-0.4701, -0.2173)
Position 2: (0,0)	(-0.1479, -0.1121)	(-0.0642, -0.1235)

The use of synthetic data allows the accurate quantification of the error in position estimation. The error (*err_pstn*) is computed as the Euclidian distance between the expected position and the estimated one, as following:

$$\text{err\_pstn} = \sqrt{(x\_real - x\_est)^2 + (y\_real - y\_est)^2} \quad (48)$$

where (x_real, y_real) is the correct position of the synthetic human model and (x_est, y_est) the estimated one, by the proposed method.

The error derived is expressed in units of meters in this work. Table II shows the error of the position estimation for each combination of models with positions 1 and 2.

Table II

The error of the position estimation for the two models at the two positions 1 and 2.

Error	**Position 1**: (-0.4,-0.1)	**Position 2**: (0,0)
Model *i*	0.1927	0.1856
Model *ii*	0.1366	0.1392

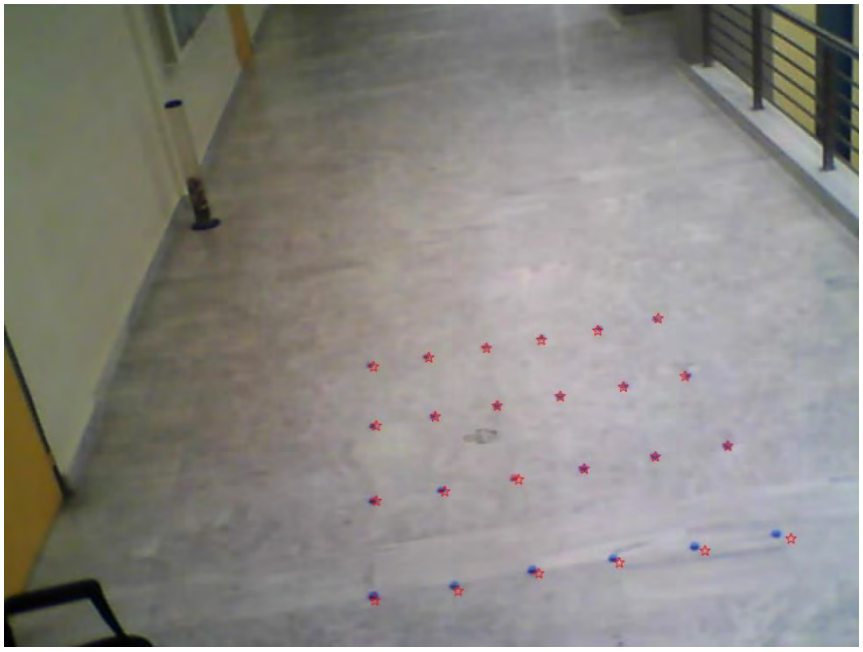
3.2.1.2. Real Data

Results from real data are presented using two different types of simple geometric objects (bottle caps and a box) and a human subject.

Simple Geometric Objects

The 24 points that were marked on the floor for the estimation of the calibration parameters were round bottle caps with diameter equal to 3.4 cm. The real world coordinates of the caps were manually measured. The positions of these simple objects were also manually identified on the frame of each camera, (because of their small size). The resulting position of each one of the 24 caps estimated using the calibration of each camera was re-projected on the corresponding frames. Thus, results are superimposed on the real frames of the three projective cameras in Fig.27.

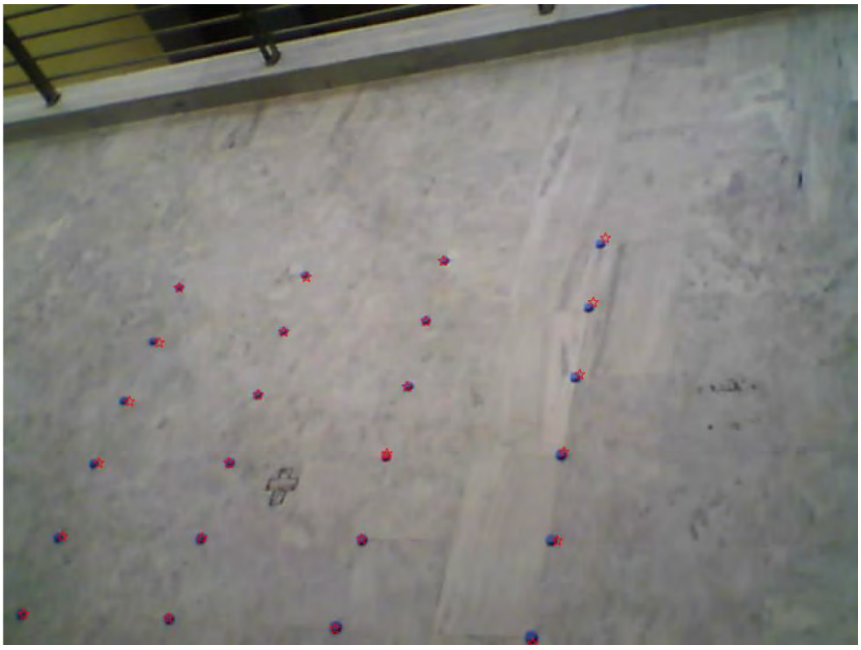
Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7



(a)

(b)

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7



(c)

Figure 27. Position estimation result plotted as red stars on the corresponding frames of the 3 projective cameras (a-c).

The resulting (estimated) real positions are plotted, along with the virtual representation of the real caps at their manually measured real positions, on an *X*-*Y* axis diagram (the *Z* axis is omitted) shown in Fig.28.

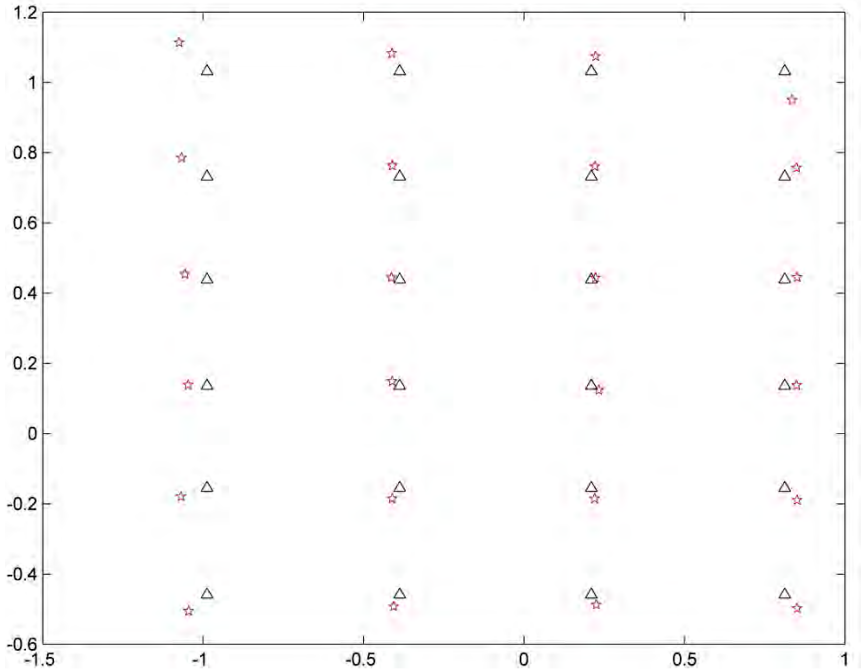


Figure 28. The resulting positions (red stars), along with the virtual representation of the real positions of the caps (blue triangles), on an *X*-*Y* axis diagram.

The plotted positions of the above diagram are also presented numerically in the following Table, along with the error of position estimation for each cap (Table

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

III). The values of this Table were used in the calculation of the average error of the position estimation.

Table III

The *X*-*Y* axis coordinates of the estimated and of the real positions of the caps (two first columns – left and right columns respectively) and the error of the real position estimation for each cap (last column) measured in meters.

Estimated positions (m)		Real positions (m)		Position Error (m)
***X* coord.**	***Y* coord.**	***X* coord.**	***Y* coord.**	
0.8352	0.9492	0.8130	1.0320	0.0857
0.8498	0.7569	0.8130	0.7320	0.0444
0.8510	0.4447	0.8130	0.4380	0.0385
0.8501	0.1377	0.8130	0.1360	0.0371
0.8515	-0.1897	0.8130	-0.1550	0.0518
0.8517	-0.4979	0.8130	-0.4590	0.0549
0.2239	1.0737	0.2100	1.0320	0.0439
0.2214	0.7607	0.2100	0.7320	0.0309
0.2232	0.4432	0.2100	0.4380	0.0142
0.2335	0.1233	0.2100	0.1360	0.0267
0.2204	-0.1854	0.2100	-0.1550	0.0322
0.2250	-0.4876	0.2100	-0.4590	0.0323
-0.4112	1.0826	-0.3870	1.0320	0.0561
-0.4093	0.7624	-0.3870	0.7320	0.0377
-0.4127	0.4439	-0.3870	0.4380	0.0263
-0.4102	0.1484	-0.3870	0.1360	0.0263
-0.4106	-0.1857	-0.3870	-0.1550	0.0387
-0.4057	-0.4928	-0.3870	-0.4590	0.0387
-1.0742	1.1138	-0.9870	1.0320	0.1196
-1.0663	0.7855	-0.9870	0.7320	0.0956
-1.0561	0.4540	-0.9870	0.4380	0.0709
-1.0458	0.1381	-0.9870	0.1360	0.0588
-1.0677	-0.1799	-0.9870	-0.1550	0.0844
-1.0446	-0.5056	-0.9870	-0.4590	0.0741

The average error of the position estimation of the 24 caps is calculated as following:

$$\frac{1}{24}\sum_{i=1}^{24}\sqrt{(x\_real_i - x\_est_i)^2 + (y\_real_i - y\_est_i)^2} \qquad (49)$$

where $(\mathbf{x\_real}, \mathbf{y\_real})$ are the real world coordinates of the set of the caps and $(\mathbf{x\_est}, \mathbf{y\_est})$ the coordinates of the estimated positions of theirs by the proposed method. The average error for the position estimation of the 24 caps is equal to 0.0508 meters.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Furthermore, the proposed algorithm was applied to another simple geometric object of the shape of a box. Due to the small size of the object, the segmentation was performed manually. The original frames, of the available projective cameras viewing the object, and the corresponding segmented ones are shown in Fig. 29 (upper and lower row respectively).

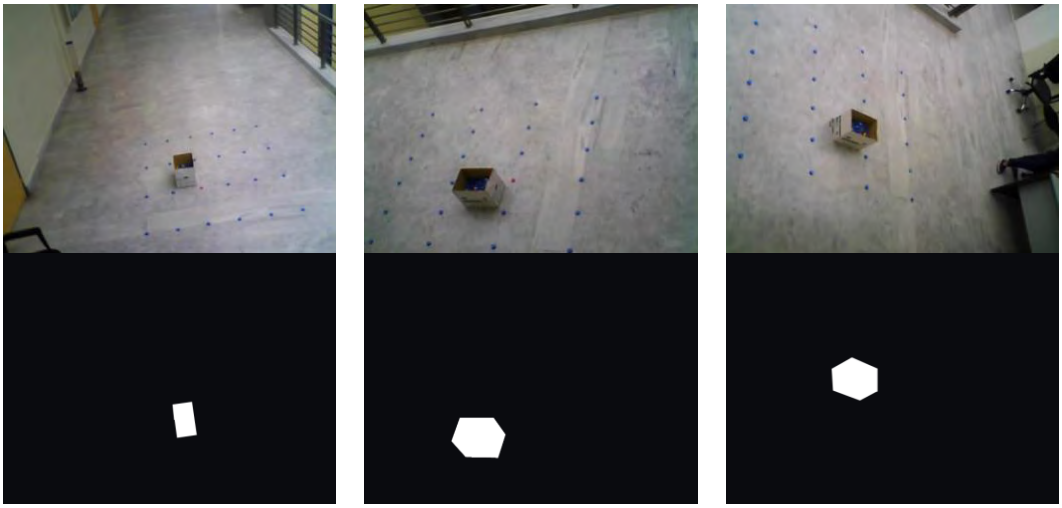


Figure 29. The original (upper) and the corresponding segmented (lower) frames containing the object of interest.

The proposed position estimation algorithm can be also applied in the case of an object that covers relatively large area on the floor, although with less accurate results if only one camera is used. The maximum Z axis coordinates of the scene vector $\mathbf{b}_c$ (as defined in Eq.(45)) resembles the vector that points to the lower point of the segmented silhouette as each camera views it. Thus, this pixel can be used to estimate the object's position on the floor. However, each camera views the object from a different angle, so applying the position estimation algorithm independently from each silhouette (viewed by the cameras) will produce different position estimation. The average of these estimations results in a more accurate position estimation of the center of mass of the object, especially if the cameras view the object from significantly different angles. In Fig. 30, each resulting position, estimated from the view of each one of the available cameras, is superimposed on the corresponding frame of the segmented silhouette of the object. The binary frames are cropped around the segmented silhouette in order to illustrate a detailed view of the position estimation, since the object covers a small area of the frame. The average of the three different position estimations is superimposed on a real world frame of the object, acquired by one of the three cameras.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

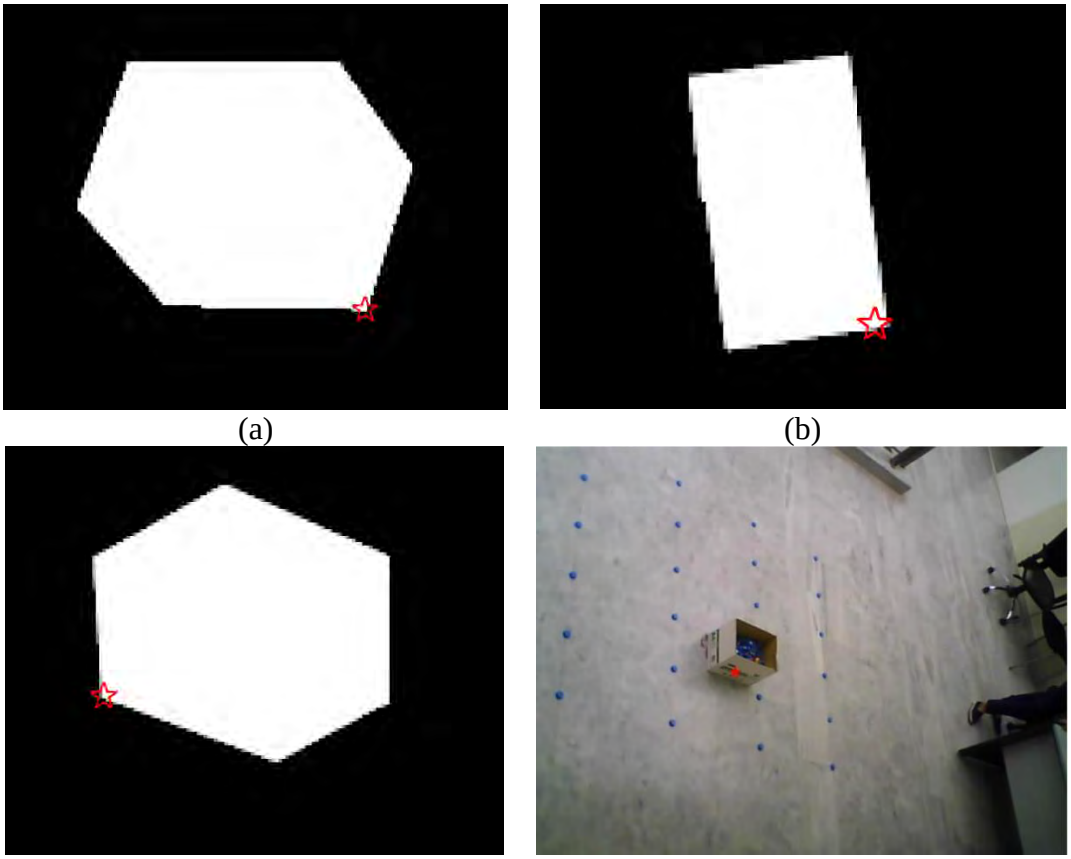


Figure 30. Position estimation from the view of each one of the available cameras plotted as red stars on the corresponding segmented frame (a-c), cropped around the segmented silhouette, since the object covers a small area of the frame. The average of the three different position estimations indicated in a real world frame of the object, acquired by one of the three cameras.

The values of the real position of the box (x_real, y_real) on *X* and *Y* axis (the *Z* axis coordinate has always the value on which the floor is) are indicated in Table IV along with the estimated position coordinates (x_est, y_est) resulted by the proposed method of position estimation.

Table IV

Comparison of the real world and of the estimated by the proposed method position of the box.

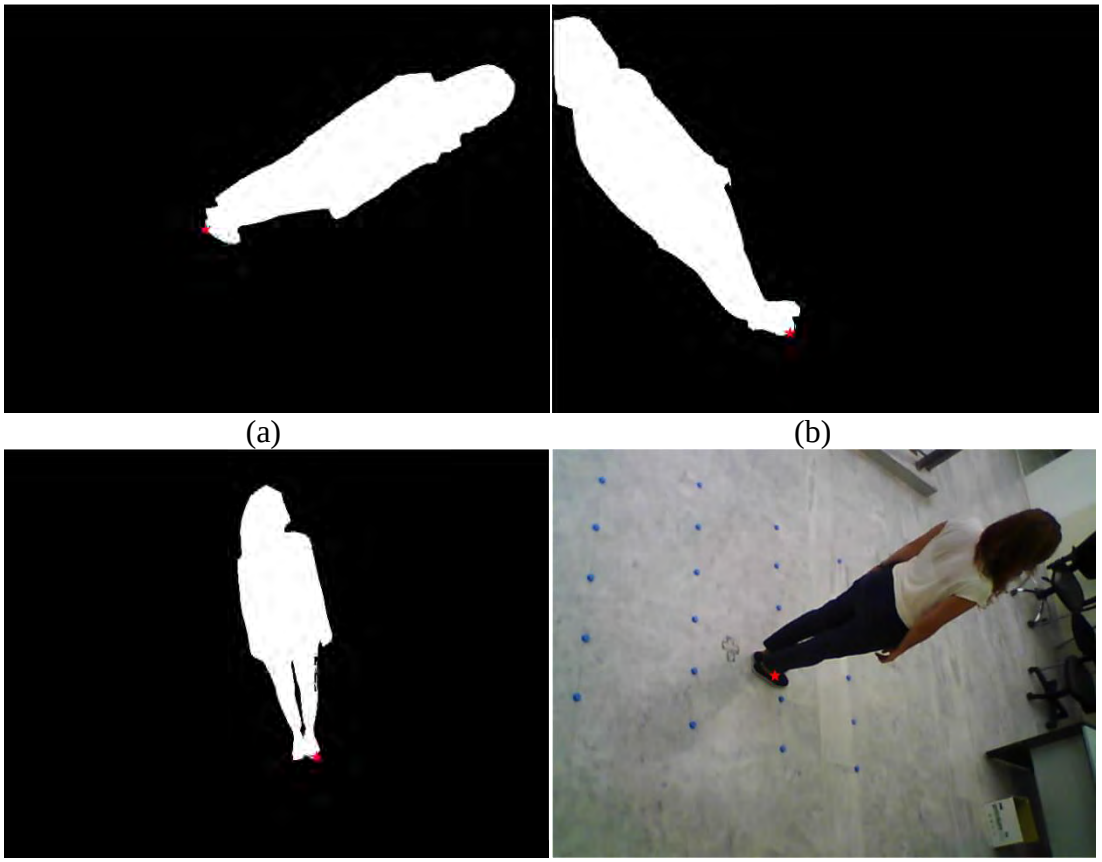
	***X* coord. (m)**	***Y* coord. (m)**
Real position	-0.1730	-0.0190
Estimated position	-0.2201	-0.0751

The error for the position estimation of the real box is equal to 0.0733 meters.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Human Model

The position estimation was also applied on real data of human model and specifically on the frames of the subject of Fig.40(a). The estimated real world position, using each available camera, was re-projected on the corresponding binary frame B_c as a red-filled star (Fig.31,a-c), in order to be visually tested. The average of the estimated real world position of each one of the available cameras is plotted as a red star in a real world frame of the human model, acquired by one of the three cameras, in Fig.31(d).



(a) (b)

(c) (d)

Figure 31. Position estimation result plotted as red-filled stars on the corresponding segmented frame (a-c). The average of the three different position estimations indicated in randomly one of the three binary frames.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

3.2.2. *Application of S2S*

The first requirement of Silhouette to shape method (S2S, commonly referred as shape from silhouette – SfS), in order to reconstruct the object of interest, is the set of frames that contain the object and are acquired by one or more cameras. In this work, both synthetic and real models were rendered or captured, respectively, and three projective cameras were mounted at specific positions in the room in order to share a common portion of real world space as their FoV. The exact position of each camera is shown in a virtual representation of the real room in Fig.8. The pixilation of each frame is 480x640. The D-Link DCS-930/L was selected as the three projective cameras.

3.2.2.1. Synthetic Data

Synthetic data are used to quantify the error of the achieved shape reconstruction, while isolating the effects of segmentation and calibration. A 3D human model, with known real world points, as explained in Subsection 3.2.1.1, is virtually placed in room at a certain position within the FoV of all the projective cameras (Fig.32). These models are in the form of triangulated surfaces. However, only the coordinates of the vertices are used for rendering the binary silhouette frames.

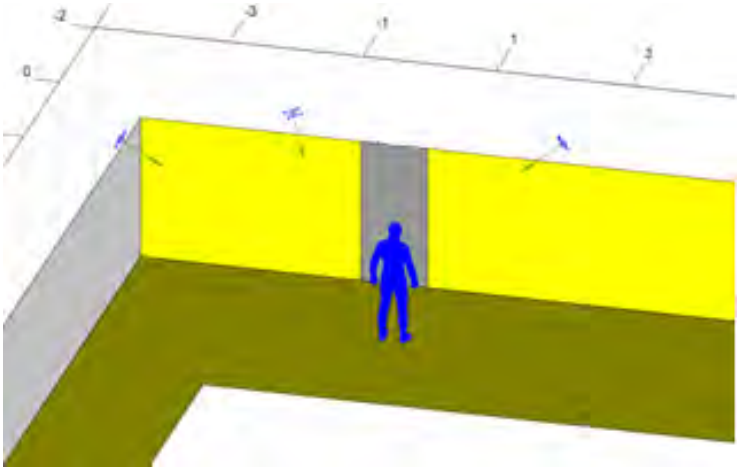


Figure 32. 3D representation of the room with the three projective cameras (with orientation indicated by normal vectors) and the synthetic 3D human model placed at a certain position in the room.

Each real world point (vertex) of the 3D human model is rendered according to calibration. More specifically, the corresponding binary frames B_c (Fig.33), that contain the silhouette are generated using functions F_c (explained in Subsection 2.4.3). Using the S2S algorithm analyzed in Subsection 2.4.3, volume V is obtained. These steps, along with an extra example of the reconstruction of synthetic model, are figured in the block diagram of Fig. 34.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Figure 33. The rendered binary frames of the projective cameras (B_c) in the case of synthetic data.

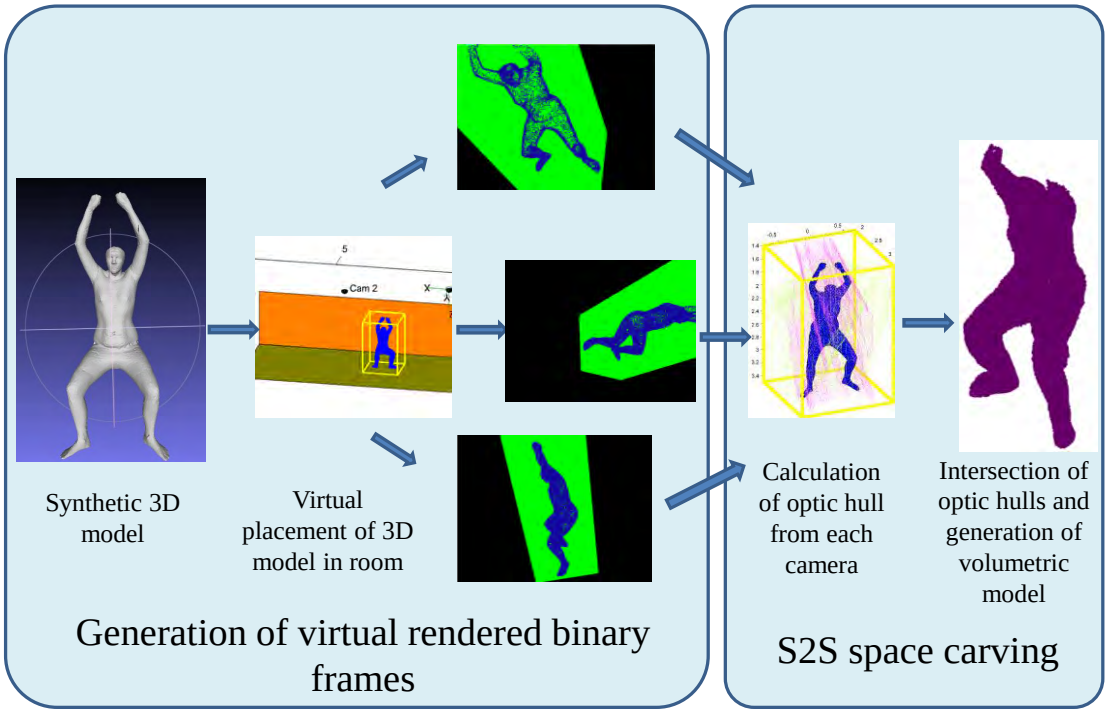


Figure 34. The steps for the volumetric reconstruction of synthetic data followed in this work.

In the case of synthetic data, the visually assessed error of the reconstructed model is only due to the S2S method. More specifically, as explained in Subsection 1.3.2, infinite views of the model may conclude to a reconstruction identical with the real model, where it is convex. However, the discrepancy between the reconstructed shape and the 3D model at the upper part of the hands is caused by the clipping of the model in two of the projective silhouettes as shown in the two upper rendered frames of Fig.34 (left side, third column) – silhouettes are more clearly illustrated in Fig.(23).

Calculation of Reconstruction Error

The use of synthetic data allows the accurate quantification of the error in volume reconstruction. The resulting optic hull is generated in the form of a volumetric model *V*. A simple linear transformation from cloud space to voxel space is applied to the points of the 3D human model in order to calulate the displacement error with respect to the 3D human model, which is in the form of point cloud. Thus, volume V_1 is generated using voxel size of δl_0=1 cm. Subsequently, the

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

reconstructed volume V is resized using nearest neighbour interpolarion to obtain equal dimensions to V_1. Finally, the Euclidean distance transform (DT) of V_1 is computed in 3 dimensions. The average error (*err*) is computed as the average distance of the non-zero voxels of V from the non-zero voxels of V_1 as following:

$$err = \frac{1}{N}\sum_{i,j,k} DT(V_1)_{ijk}\ ,\ N = \#\left\{(i,j,k): V_{ijk} > 0\right\} \qquad (50)$$

The error derived is expressed in units of δl_0 (which is set equal to 1 cm for the results presented in this work). Table V shows the average error of the reconstructed volume, for different combinations of the participating projective cameras (1 indicates camera participation).

Table V

Comparison of the average error, for different combinations of the participating projective cameras.

P1	P2	P3	Average Error (cm)
1	1	0	4.669
1	0	1	3.796
0	1	1	3.558
1	1	1	2.961

Fig.35 illustrates the reconstructed volume V with the ground truth model superimposed. The accuracy of the reconstruction can be visually assessed.

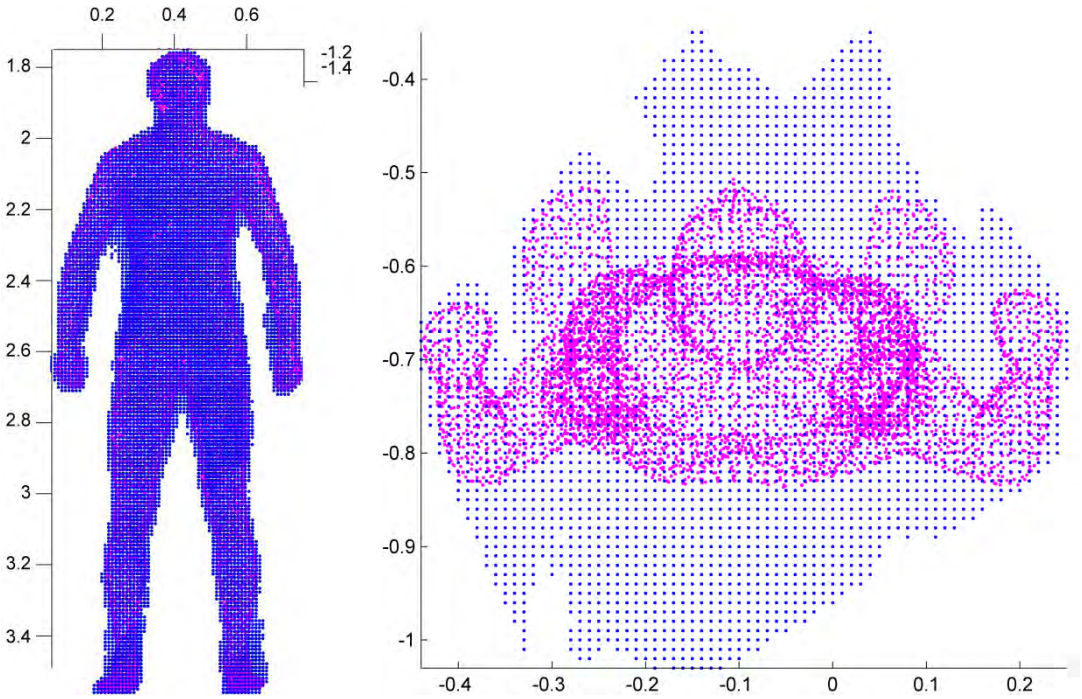


Figure 35. The final reconstructed model using all available cameras and the ground truth model, shown in blue and magenda respectively. En-face (left) and top-down view (right).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

3.2.2.2. Real Data

Results from real data are presented using a simple geometric object and several human subjects, using the three aforementioned projective cameras, calibrated as described in Subsection 2.3.

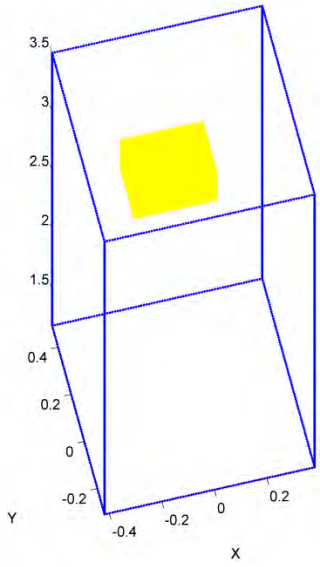
Simple Geometric Object

The proposed algorithm is applied to reconstruct the shape of a simple object (box) that is placed at a certain position on the floor. The box was imaged by the three available projective cameras. The original frames and the corresponding segmented ones are shown in Fig. 29. Due to the small size of the object and to isolate the results from the effects of possible segmentation inaccuracies, the segmentation was performed manually. The segmented frames along with the cube *C*, which is large enough to include the whole object (explained in Subsection 2.4.3), are shown for each camera in Fig.36.

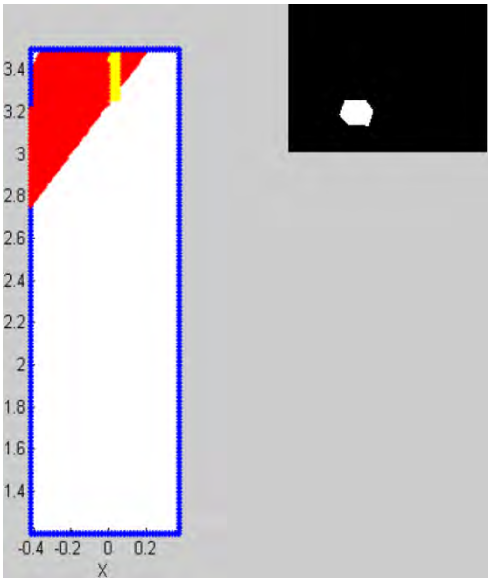
Figure 36. The segmented frames along with the cube *C*. The cube *C* is indicated on the red channel of the frame.

The steps of the real box reconstruction are shown in Fig.37. Firstly, the virtual representation of the real object is shown in real world coordinates within cube *C*. In the experimental setup of this work, the level of the ceiling is set to zero and *Z* axis is positive towards the floor of the room. Therefore, the lower side of the box lies on the plane defined by z_0 =3.5 m (equal to room height). Then, the generalized cones (visual hull of each camera) that are generated by the images of Fig.36 are projected inside the VOI (Fig.37, b, c and d). The B_c frames that contribute to the creation of each visual hull are also shown (on the right side of (b)-(d) frames of Fig.37). Finally, the intersection of the visual hulls is shown in real world coordinates superimposed on the real object that represents the ground truth (Fig.37,d).

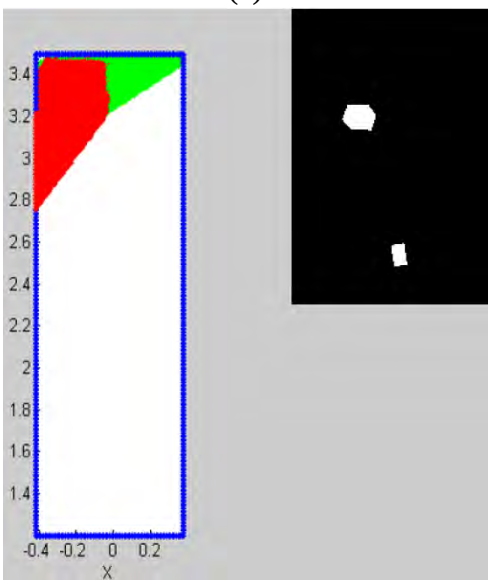
Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7



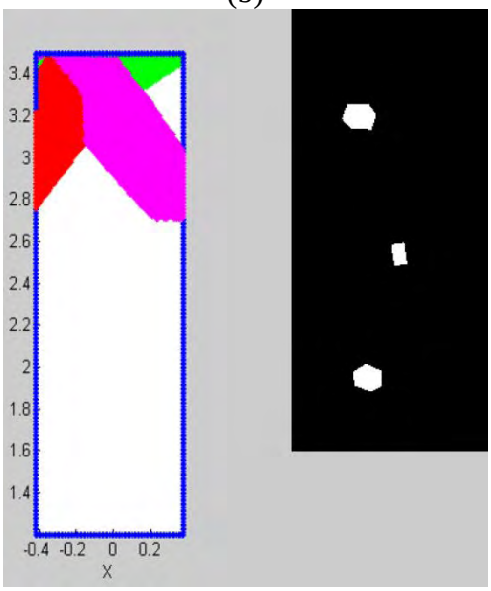
(a)



(b)

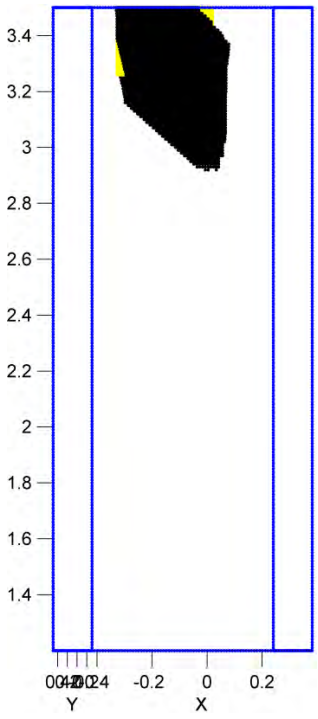


(c)



(d)

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7



(e)

Figure 37. The virtual representation of the real object (yellow box) and the cube *C*, in real world coordinates (a), the generalized cones along with corresponding B_c frames (b - d) and the resulting reconstructed shape (black) superimposed on the real object (e). The real world axes are also indicated.

The intersection of the visual hulls concludes to the 3D reconstruction of the viewed object. A better view of the resulting volumetric reconstructed object (blue) superimposed on the real object (magenta) is plotted in Fig.38. The discrepancy between the reconstructed volume and the 3D real object because of the clipping of the reconstructed shape at the right-viewed part of the box is caused by the imperfection of the calibration. The effect of the calibration on the reconstruction is analyzed in the next Subsection.

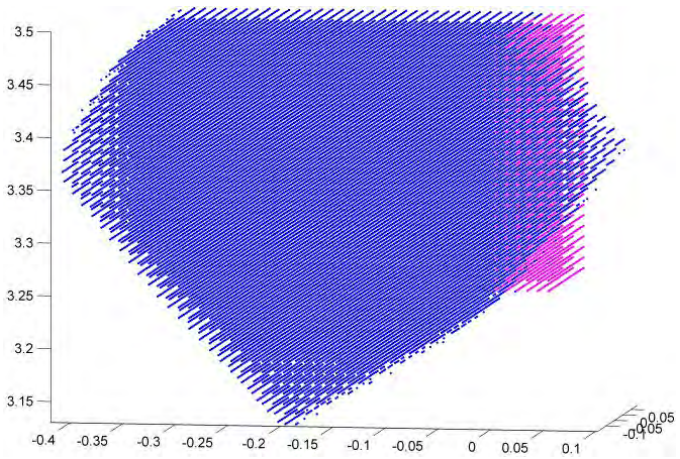


Figure 38. The resulting volumetric reconstructed object (blue) superimposed on the real object-box (magenta).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

The reconstruction error of the specific simple geometric object (box) can be calculated since its dimensions are easily measured. Table VI shows the average error of the reconstructed volume, for different combinations of the participating projective cameras (1 indicates camera participation).

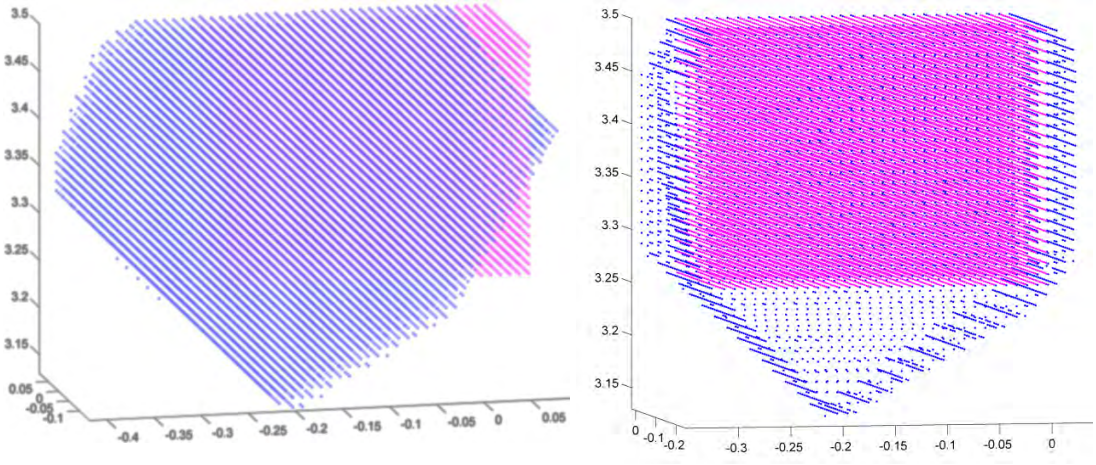
Table VI

Comparison of the average error for real data, for different combinations of the participating projective cameras.

P1	P2	P3	Average Error (cm)
1	1	0	5.270
1	0	1	6.328
0	1	1	5.285
1	1	1	4.836

The effect of calibration on reconstruction error

In order to test the effect of the calibration on the reconstruction, a synthetic geometric object of the same shape and dimensions with the real box was constructed. The S2S algorithm was separately applied to the frames of the real box and the rendered frames of the synthetic box. The resulting reconstructed shapes of the two aforementioned cases are depicted in Fig.39.



(a) (b)

Figure 39. . The reconstructed shapes of the real (a) and the synthetic (b) box of the same dimensions. Blue dots indicate the reconstructed box, whereas magenta dots indicate the ground truth box.

The difference in the accuracy of reconstruction is evident. In the case of synthetic box (Fig.39,b), the reconstructed object correctly encloses the box and does not intersect with it, whereas its volume is not exceedingly large. On the other hand, the reconstructed volume using the real box (thus suffering camera calibration errors) is of lower quality: it does not enclose the whole box and its volume is clearly larger than the one in the synthetic case. The average reconstruction error validates this visual assessment. Table VII shows the average error of the two reconstructed volumes of the case of real and synthetic data of Fig.39.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Table VII

Comparison of the average error for real and synthetic data of the same object.

Input Data	Average Error (cm)
Real Box	6.15
Synthetic Box	2.11

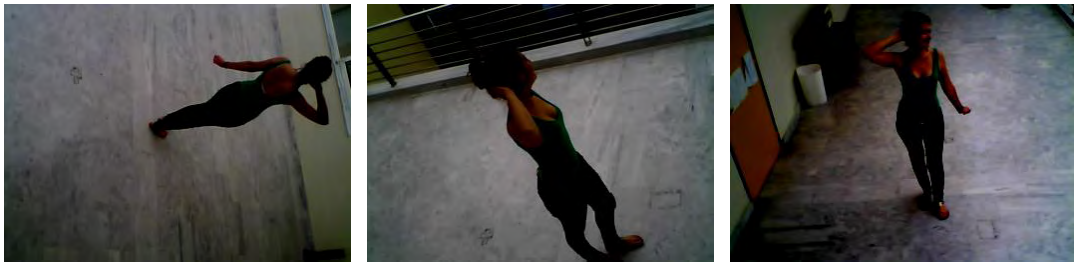
Therefore, calibration greatly affects the reconstruction of volumetric objects through S2S algorithm.

Human Model

Two human subjects were asked to stand in the room in three different postures and at certain positions within the FoV of all the projective cameras. Images were acquired from all the cameras before and after the placement of the subjects. Afterwards, the corresponding binary frames B_c (Fig.40), that contain the silhouette, where produced by a simple segmentation method, as explained in Subsection 2.4.1. Manual segmentation was employed for the second subject (Fig. 40,e and f) due to poor light conditions.

(a)

(b)



(c)

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

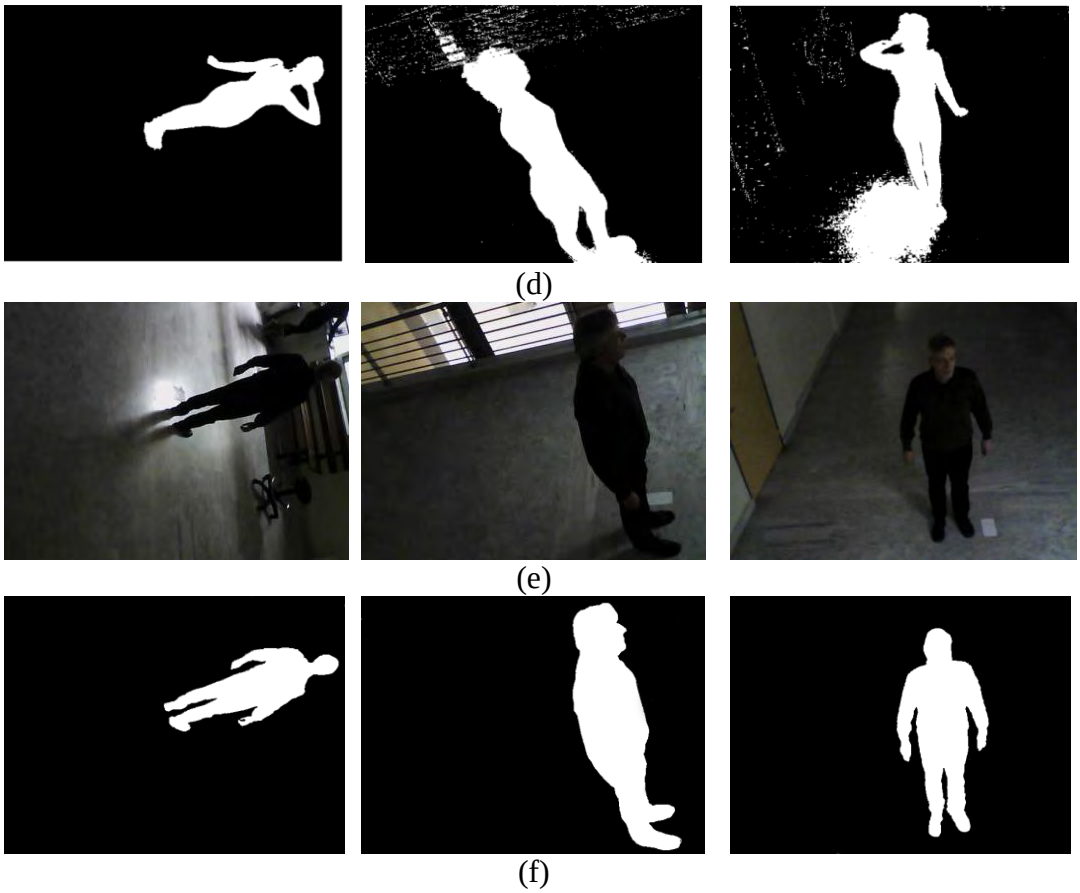
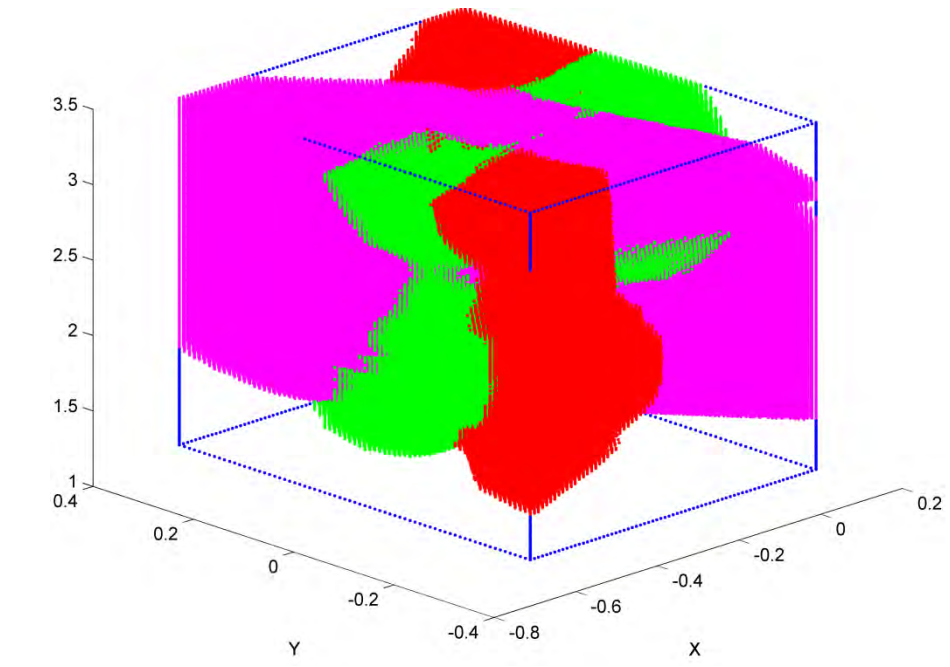


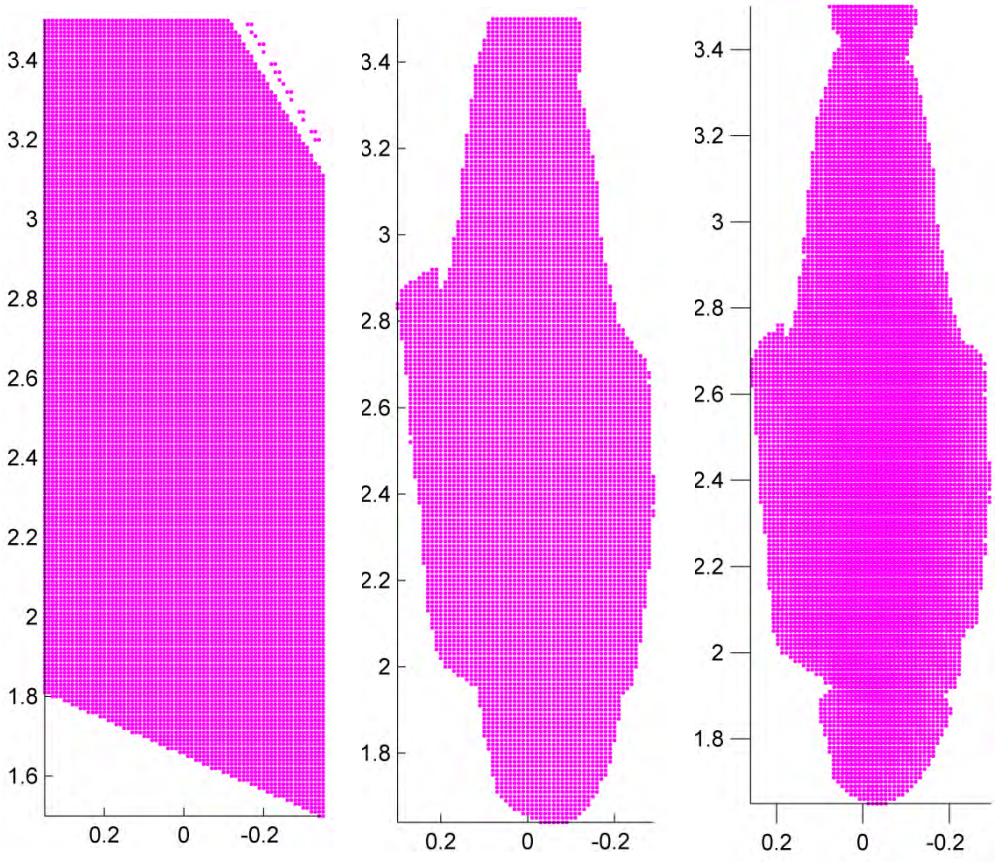
Figure 40. The original frames (a,c,e) and the corresponding segmented frames (b,d,f) for two different persons, in three different postures (a, c and e), acquired by the three projective cameras. The frames of the second person (e) have been manually segmented.

The volumetric models *V* of the human subjects were produced for each posture using the S2S algorithm of Subsection 2.4.3. The generalized cones produced by the corresponding camera's binary silhouette is plotted in different color in cube *C* of Fig. 41(a) for the first pose of the first subject of Fig.40(a). The sequence that the projections took place was by color: red, green and lastly the magenta. The initial visual hull of the subject (which is identical to the generalized cone of the first projective camera – the red cone), is shown in Fig.41(b). The intersection of the red cone with the green cone resulted in the visual hull of Fig.41(c). The final reconstructed volume, created by the intersection of the two visual hulls of (b) and (c), is depicted in Fig.41(d).

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7



(a)

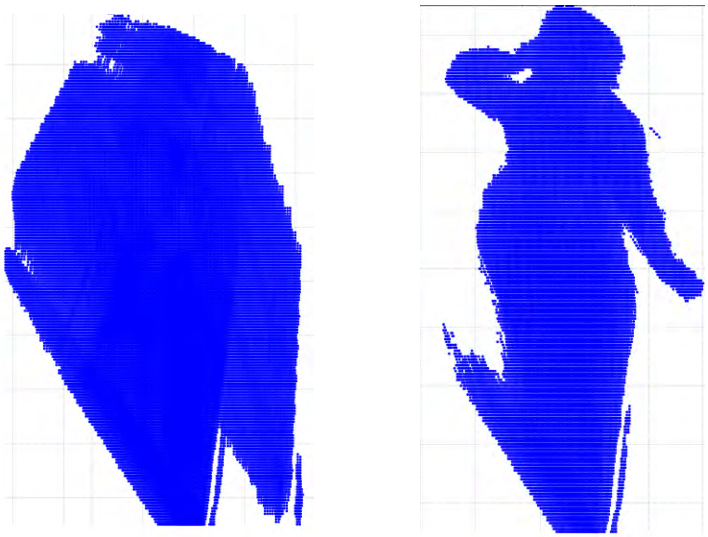


(b) (c) (d)

Figure 41. The three generalized cones of the three projective cameras shown in the cube *C* of (a). The progression of the resulting volumetric model of the subject of Fig. 40(a).

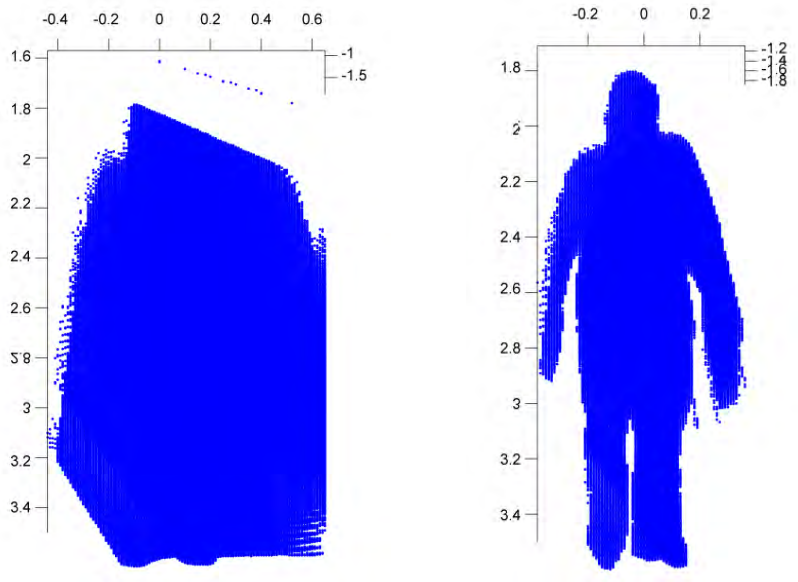
Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Furthermore, Fig.42 and Fig.43 show the progression of the resulting volumetric models of the subjects of Fig 40(c) and (e), respectively, using (a) the two projective cameras and (b) all the available projective cameras.



(a) (b)

Figure 42. Volumetric model of the second posture of the first person (of Fig.40,c), generated using: (a) the frames of two projective cameras and (b) including additionally the frame of the third projective camera.



(a) (b)

Figure 43. Volumetric model of the second person (of Fig.40,e), generated using: (a) the frames of two projective cameras and (b) including additionally the frame of the third projective camera.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

4. Conclusion and Further Work

Two techniques of perspective calibration [1], [2] are reviewed using linear and non-linear estimation of the cameras' projection matrices, while several mathematical variations on them are proposed. Calibration results are utilized for the implementation of two computer vision applications: real-time position estimation and volumetric reconstruction from silhouettes.

The required initial step of segmentation of objects of interest has been discussed. The proposed position estimation algorithm works efficiently for both simple geometric object's and human model's position estimation. The algorithm can also be applied in the case of an object that covers relatively large area on the floor, although with more accurate results if more than one camera is used. Results are presented for the case of three cameras view and the calculated error of the method is significantly low.

Moreover, an algorithm is described for the reconstruction of shape from multiple silhouettes, acquired from three projective cameras. Initial results from reconstructing shapes of real human subjects are assessed visually and appear promising. The proposed algorithm is robust enough to work successfully, despite the quality of the segmentation. However, the calibration method affects strongly our method. The effect of calibration in reconstruction error is examined and analyzed. In order for the results to be ameliorated, the accuracy of the calibration method should be improved.

The calibration of perspective camera, as well as the applications of computer vision have been implemented using Matlab. Reconstruction time for the reconstruction algorithm was less than 1 second, for 3 projective using a volume of 1.5×1.5×2.2 meters as VOI, with an elementary cube element δV of 1×1×1 cm and executed on an Intel(R) Core i5-2430 CPU @ 2.40 GHz Laptop with 4 GB Ram, under Windows 7 Home Premium. The experimental setup of the work consists of three, mounted at specific positions, projective cameras D-Link DCS-930/L. The pixilation of each frame is 480x640.

Future work includes extending the position estimation and volume reconstruction using views from multiple fisheye cameras, or heterogeneous network of cameras. Using more accurate camera calibration algorithm can also be investigated. Moreover, methods for reducing the dependence of the S2S algorithm from the accuracy of the calibration can be considered. Furthermore, the feasibility of continuous shape reconstruction, at different positions and times along any trajectory is a useful and ambitious task. The exploitation of these algorithms, also require the use of shape descriptors in 3D in order to identify posture. Finally the possibility of volume reconstruction from objects under different poses needs be investigated, perhaps using more information than the binary silhouettes.

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

5. References

1. Horn, B. K. (2000). Tsai's camera calibration method revisited. Online: http://people. csail. mit. edu/bkph/articles/Tsai_Revisited. pdf.
2. Z. Zhang, "Camera Calibration", Chapter 2, pages 4-43, in G. Medioni and S.B. Kang, eds., Emerging Topics in Computer Vision, Prentice Hall Professional Technical Reference, 2004.
3. Mohr, R., & Triggs, B. (1996, July). Projective geometry for image analysis. In *XVIIIth International Symposium on Photogrammetry & Remote Sensing (ISPRS'96)*.
4. K. Kutulakos and S. Seitz "A theory of shape by space carving," International Journal of Computer Vision, vol. 38(3), pp.199–218, 2000
5. M. Potmesil, "Generating octree models of 3D objects from their silhouettes in a sequence of images," CVGIP, vol. 40, pp. 1-29, 1987.
6. T. B. Moeslund and E. Granum, A Survey of Computer Vision-Based Human Motion Capture, Computer Vision and Image Understanding 81, 231–268 (2001).
7. Hughes, J. F., Van Dam, A., Foley, J. D., & Feiner, S. K. (2013). *Computer graphics: principles and practice*. Pearson Education. – par.11.2.2.
8. Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, *58*(15-16), 1-35. – Subsection 5.1
9. Samtaney, R. (1999). A method to solve interior and exterior camera calibration parameters for image resection.
10. Gallian, J. (2006). Abstract Algebra. 6e. Boston, Mass.: Houghton Mifflin.
11. Mebius, J. E. (2007). Derivation of the Euler-Rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations. arXiv preprint math/0701759.
12. R. Tsai, "A versatile camera calibration technique for High-Accuracy 3D Machine Vision Metrology using Off-the –self TV Cameras and Lenses," IEEE J of Robotics and Automation, vol. 3(4), pp. 323-344, 1987.
13. J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, vol. 9(1), pp. 112-147, 1998.
14. Hasler, N., Ackermann H., Rosenhahn B., Thormahlen T. Seidel H.P.: Multilinear pose and body shape estimation of dressed subjects from image sets. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010), pp. 1823–1830. (2010).
15. http://resources.mpi-inf.mpg.de/scandb/

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7

Institutional Repository - Library & Information Centre - University of Thessaly
09/12/2017 12:59:13 EET - 137.108.70.7